



**TUGAS AKHIR - KI091391**

# **OPTIMASI K-MEANS DENGAN ARTIFICIAL BEE COLONY UNTUK PENGELOMPOKAN RESOLUSI TAHUN BARU DARI DATA TWITTER**

**RIZKI NOVIANTO**  
**NRP 5109 100 143**

Dosen Pembimbing I  
Dr.Eng. Chastine Fatichah, S.Kom., M.Kom.

Dosen Pembimbing II  
Diana Purwitasari, S.Kom., M.Sc.

**JURUSAN TEKNIK INFORMATIKA**  
**Fakultas Teknologi Informasi**  
**Institut Teknologi Sepuluh Nopember**  
**Surabaya 201**





**TUGAS AKHIR - KI091391**

# **OPTIMASI K-MEANS DENGAN ARTIFICIAL BEE COLONY UNTUK PENGELOMPOKAN RESOLUSI TAHUN BARU DARI DATA TWITTER**

**RIZKI NOVIANTO**  
**NRP 5109 100 143**

**Dosen Pembimbing I**  
**Dr.Eng. Chastine Fatichah, S.Kom., M.Kom.**

**Dosen Pembimbing II**  
**Diana Purwitasari, S.Kom., M.Sc.**

**JURUSAN TEKNIK INFORMATIKA**  
**FAKULTAS TEKNOLOGI INFORMASI**  
**INSTITUT TEKNOLOGI SEPULUH NOPEMBER**  
**SURABAYA 2017**

***(Halaman ini sengaja dikosongkan)***



**UNDERGRADUATE THESES - KI091391**

# **K-MEANS OPTIMIZATION USING ARTIFICIAL BEE COLONY FOR CLASSIFICATING NEW YEAR RESOLUTION DATA FROM TWITTER**

**RIZKI NOVIANTO**  
**NRP 5109 100 143**

**First Advisor**  
**Dr.Eng. Chastine Fatichah, S.Kom., M.Kom.**

**Second Advisor**  
**Diana Purwitasari, S.Kom., M.Sc.**

**DEPARTMENT OF INFORMATICS**  
**FACULTY OF INFORMATION TECHNOLOGY**  
**SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY**  
**SURABAYA 2017**

***(Halaman ini sengaja dikosongkan)***

## LEMBAR PENGESAHAN

### OPTIMASI K-MEANS DENGAN ARTIFICIAL BEE COLONY UNTUK PENGELOMPOKAN RESOLUSI TAHUN BARU DARI DATA TWITTER

#### TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada  
Bidang Studi Komputasi Cerdas Visual  
Program Studi S-1 Jurusan Teknik Informatika  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember

Oleh

**RIZKI NOVIANTO**  
**NRP : 5109 100 143**

Disetujui oleh Dosen Pembimbing Tugas Akhir:

1. Dr.Eng. Chastine Fatichah, S.Kom., M.Kom.  
NIP: 197512202001122002
2. Diana Purwitasari, S.Kom., M.Sc.  
NIP: 197804102003122001



**SURABAYA**  
**MEI, 2017**

***(Halaman ini sengaja dikosongkan)***



# OPTIMASI K-MEANS DENGAN ARTIFICIAL BEE COLONY UNTUK PENGELOMPOKAN RESOLUSI TAHUN BARU DARI DATA TWITTER

Nama Mahasiswa : RIZKI NOVIANTO  
NRP : 5109100143  
Jurusan : Teknik Informatika FTIF-ITS  
Dosen Pembimbing 1 : Dr.Eng. Chastine Fatichah, S.Kom.,  
M.Kom.  
Dosen Pembimbing 2 : Diana Purwitasari, S.Kom., M.Sc.

## Abstrak

Media sosial khususnya *Twitter* merupakan suatu sumber data yang kaya, sehingga membuka kesempatan untuk penerapan *data mining* untuk menghasilkan suatu data yang dapat dimanfaatkan untuk berbagai kepentingan. Termasuk dalam berbagai data yang ada ialah resolusi tahun baru yang dapat digeneralisasi menjadi beberapa kelompok.

Salah satu metode pengelompokan data yang umum digunakan adalah *K-means clustering* (KM). Meski demikian, metode *K-means* memiliki beberapa kelemahan, salah satunya yaitu dalam penentuan titik awal *centroid*.

*Artificial Bee Colony* (ABC) merupakan metode yang terinspirasi dari cara lebah mencari makanan untuk kelompoknya. Metode ABC dikenal dapat menghindari permasalahan lokal optima, yang umumnya terjadi pada penggunaan *K-means* karena kelemahan dalam penetapan titik *centroid*.

Penggunaan kombinasi algoritma *K-means* dengan ABC dapat menjadi solusi dalam permasalahan pengelompokan pada *data mining*. Tugas Akhir ini bertujuan untuk mencari tahu seberapa efektif penggunaan gabungan *K-means* dengan ABC (ABCKM) dalam pengelompokan data resolusi tahun baru dari *Twitter*.

**Kata Kunci:** *Resolusi Tahun Baru, Data Mining, K-Means, Artificial Bee Colony, Clustering.*

***(Halaman ini sengaja dikosongkan)***

# **K-MEANS OPTIMIZATION USING ARTIFICIAL BEE COLONY FOR CLASSIFICATION OF NEW YEAR RESOLUTION DATA FROM TWITTER**

**Student's Name** : RIZKI NOVIANTO  
**Student's ID** : 5109 100 143  
**Department** : Teknik Informatika FTIF-ITS  
**First Advisor** : Dr.Eng. Chastine Fatichah, S.Kom., M.Kom.  
**Second Advisor** : Diana Purwitasari, S.Kom., M.Sc.

## **Abstract**

*Social Media, especially Twitter is a rich source of data, providing the chance to implement data mining to get some useful data for many kinds of interest. One kind of those data is new year resolution which could be generalized as groups.*

*A commonly used way to divide data into groups is K-means clustering. However, K-means method itself have some weakness, one of them lies in the selection of the initial centroid.*

*Artificial Bee Colony (ABC) is a method that inspired by how bees behave in search of food for its colony. ABC method is known for the ability to avoid local optima problem, often happens in the usage of K-means which caused by miss in creation of the initial centroid.*

*Using both K-means and Artificial Bee Colony could potentially be a solution to grouping problems in data mining. The main purpose of this work is to test how effective the combination of K-means and ABC (ABCKM) in grouping the new year resolution data from Twitter.*

**Keywords:** *New Year Resolution, Data Mining, K-Means, Artificial Bee Colony, Clustering.*

*(Halaman ini sengaja dikosongkan)*

## KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillahirabbil'alamin, segala puji bagi Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul:

### **Optimasi *K-means* dengan *Artificial Bee Colony* untuk Pengelompokan Resolusi Tahun Baru dari Data *Twitter***

Selesainya Tugas Akhir ini tidak lepas dari bantuan dan dukungan berbagai pihak. Sehingga pada kesempatan ini penulis mengucapkan terima kasih kepada :

1. Dr.Eng. Chastine Fatichah, S.Kom., M.Kom., selaku pembimbing I yang telah membantu, membimbing, serta memberikan banyak masukan kepada penulis dalam menyelesaikan Tugas Akhir ini.
2. Diana Purwitasari, S.Kom., M.Sc., selaku pembimbing II sekaligus dosen wali selama 7 tahun yang telah memberikan banyak nasihat, arahan, dan bantuan sehingga penulis dapat menyelesaikan Tugas Akhir ini, dan seluruh bantuan dalam menyelesaikan masa perkuliahan penulis.
3. Segenap dosen dan karyawan jurusan Teknik Informatika ITS atas ilmu dan pengalaman yang telah diberikan selama penulis menjalani masa studi di jurusan Teknik Informatika ITS.
4. Keluarga khususnya kedua orang tua atas dukungan moral, material, doa, dan dorongan yang terus-menerus agar dapat menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa dalam pembuatan Tugas Akhir ini masih terdapat banyak kekurangan. Sehingga dengan kerendahan hati, penulis mengharapkan rasa maklum, juga kritik dan saran dari pembaca untuk perbaikan ke depan.

Surabaya, Juni 2017  
Rizki Novianto

***(Halaman ini sengaja dikosongkan)***

## DAFTAR ISI

LEMBAR PENGESAHAN.....	v
ABSTRAK .....	vii
<i>ABSTRACT</i> .....	ix
KATA PENGANTAR .....	xi
DAFTAR ISI .....	xiii
DAFTAR GAMBAR .....	xv
DAFTAR TABEL .....	xvii
DAFTAR KODE SUMBER .....	xix
DAFTAR NOTASI .....	xxi
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah .....	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Tugas Akhir.....	4
1.5 Manfaat.....	4
1.6 Metodologi .....	5
1.6.1 Penyusunan Proposal Tugas Akhir.....	5
1.6.2 Studi Literatur.....	5
1.6.3 Desain dan Perancangan Sistem.....	5
1.6.4 Implementasi Perangkat Lunak .....	6
1.6.5 Pengujian dan Evaluasi .....	6
1.7 Sistematika Penulisan Laporan Tugas Akhir.....	7
BAB II TINJAUAN PUSTAKA.....	9
2.1 <i>K-means</i> .....	9
2.2 <i>Artificial Bee Colony</i> .....	11
2.3 <i>Term Frequency – Inverse Domain Frequency</i> .....	14
2.4 <i>Latent Semantic Analysis</i> .....	15
2.5 <i>Silhouette</i> .....	16
BAB III DESAIN DAN PERANCANGAN PERANGKAT LUNAK .....	29
3.1 Ekstraksi Data dan Preprocessing .....	20

3.2 Analisis Performa dengan Silhouette .....	21
3.3 Aplikasi Metode ABCKM.....	22
BAB IV IMPLEMENTASI.....	29
4.1 Lingkungan Implementasi .....	29
4.2 Implementasi.....	29
4.2.1 Implementasi Tahap Preprocessing.....	29
4.2.2 Implementasi Reduksi Dimensi LSA .....	32
4.2.3 Implementasi Inisialisasi Sumber Makanan .....	33
4.2.4 Implementasi Optimasi Inisialisasi Sumber Makanan...	34
4.2.5 Implementasi <i>ABCKM</i> pada Fase <i>Employed Bee</i> .....	37
4.2.6 Implementasi <i>ABCKM</i> pada Fase <i>Onlooker Bee</i> .....	39
4.2.7 Implementasi <i>ABCKM</i> pada Fase <i>Scout Bee</i> .....	41
4.2.8 Implementasi Klasifikasi <i>Dataset</i> Menggunakan Sumber Makanan Terbaik.....	42
BAB V HASIL UJI COBA DAN EVALUASI.....	45
5.1 Lingkungan Pengujian .....	45
5.2 Skenario Uji Coba.....	45
5.3 Uji Kinerja .....	47
BAB VI KESIMPULAN DAN SARAN.....	53
6.1 Kesimpulan .....	53
6.2 Saran .....	54
Daftar Pustaka .....	54
Lampiran A Sampel Akhir Hasil Uji Coba .....	54
Lampiran B Sampel Tahapan Proses Uji Coba .....	63
BIODATA PENULIS.....	77



## DAFTAR GAMBAR

Gambar 2.1 Diagram Alir <i>Artificial Bee Colony</i> .....	12
Gambar 3.1 Diagram Alir Keseluruhan Sistem.....	19
Gambar 3.2 Diagram Alir ABCKM Sederhana .....	23
Gambar 3.3 Diagram Alir Fase Inisialisasi .....	23
Gambar 3.4 Diagram Alir Fase <i>Employed Bee</i> .....	24
Gambar 3.5 Diagram Alir Fase <i>Onlooker Bee</i> .....	25
Gambar 3.6 Diagram Alir Fase <i>Scout Bee</i> .....	26

*(Halaman ini sengaja dikosongkan)*

## DAFTAR TABEL

Tabel 2.1 Interpretasi Nilai <i>Silhouette</i> .....	16
Tabel 5.1 Uji Perbandingan Besaran Dimensi LSA.....	46
Tabel 5.2 Nilai <i>Fitness</i> Hasil Uji ABCKM.....	48
Tabel 5.3 Nilai Akurasi <i>Rand Measure</i> Hasil Uji ABCKM...	49
Tabel 5.4 Nilai <i>Precision</i> Hasil Uji ABCKM .....	50
Tabel 5.5 Nilai <i>Recall</i> Hasil Uji ABCKM .....	50
Tabel 5.6 Hasil Uji ABCKM dengan Seleksi Data .....	51
Tabel Lampiran A 1 Sumber Makanan Terbaik Uji Coba dengan SN=20, Iterasi=25, Percobaan ke-1 .....	57
Tabel Lampiran A 2 Klasifikasi Uji Coba dengan SN=20, Iterasi=25, Percobaan ke-1 .....	57
Tabel Lampiran A 3 <i>Confusion Matrix</i> Uji Coba dengan SN=20, Iterasi=25, Percobaan ke-1.....	58
Tabel Lampiran A 4 Perhitungan Akurasi Uji Coba dengan SN=20, Iterasi=25, Percobaan ke-1.....	58
Tabel Lampiran A 5 Sumber Makanan Terbaik Uji Coba dengan SN=20, Iterasi=25, Percobaan ke-2 .....	59
Tabel Lampiran A 6 Klasifikasi Uji Coba dengan SN=20, Iterasi=25, Percobaan ke-2.....	59
Tabel Lampiran A 7 <i>Confusion Matrix</i> Uji Coba dengan SN=20, Iterasi=25, Percobaan ke-2.....	59
Tabel Lampiran A 8 Perhitungan Akurasi Uji Coba dengan SN=20, Iterasi=25, Percobaan ke-2.....	60
Tabel Lampiran A 9 Sumber Makanan Terbaik Uji Coba dengan SN=50, Iterasi=50, Percobaan ke-1 .....	60
Tabel Lampiran A 10 Klasifikasi Uji Coba dengan SN=50, Iterasi=50, Percobaan ke-1 .....	61
Tabel Lampiran A 11 <i>Confusion Matrix</i> Uji Coba dengan SN=50, Iterasi=50, Percobaan ke-1.....	61
Tabel Lampiran A 12 Perhitungan Akurasi Uji Coba dengan SN=50, Iterasi=50, Percobaan ke-1 .....	61

Tabel Lampiran A 13 Hasil Akhir Uji Coba dengan SN=20, Iterasi=25 .....	62
Tabel Lampiran A 14 Hasil Akhir Uji Coba dengan SN=30, Iterasi=25 .....	62
Tabel Lampiran A 15 Hasil Akhir Uji Coba dengan SN=50, Iterasi=50 .....	62
Tabel Lampiran A 16 Hasil Akhir Uji Coba dengan SN=100, Iterasi=100 .....	62
Tabel Lampiran B 1 Data Setelah TF-IDF dan LSA .....	64
Tabel Lampiran B 2 Hasil Inisialisasi .....	64
Tabel Lampiran B 3 Hasil Optimasi Inisialisasi .....	66
Tabel Lampiran B 4 Hasil Fase <i>Employed Bee</i> untuk Iterasi ke-1 .....	68
Tabel Lampiran B 5 Hasil Fase <i>Onlooker Bee</i> untuk Iterasi ke-1 .....	71
Tabel Lampiran B 6 Hasil Akhir Sumber Makanan Terbaik .....	75

## DAFTAR KODE SUMBER

Kode Sumber 4.1 Implementasi Preprocessing.....	30
Kode Sumber 4.2 Implementasi Reduksi Dimensi LSA .....	32
Kode Sumber 4.3 Implementasi Inisialisasi Sumber Makanan .....	33
Kode Sumber 4.4 Implementasi Optimasi dengan <i>K-means</i> .....	34
Kode Sumber 4.5 Implementasi ABCKM pada Fase <i>Employed Bee</i> .....	37
Kode Sumber 4.6 Implementasi <i>Greedy Selection</i> pada Fase <i>Employed Bee</i> .....	38
Kode Sumber 4.7 Implementasi ABCKM pada Fase <i>Onlooker</i> <i>Bee</i> .....	39
Kode Sumber 4.8 Implementasi ABCKM pada Fase <i>Scout</i> <i>Bee</i> .....	41
Kode Sumber 4.9 Implementasi Klasifikasi <i>Dataset</i> .....	42

***(Halaman ini sengaja dikosongkan)***

## DAFTAR NOTASI

$SN$	Jumlah sumber makanan
$D$	Jumlah dimensi data
$K$	Jumlah <i>cluster</i>
$k$	$\{1, 2, \dots, K\}$
$i$	$\{1, 2, \dots, SN\}$
$j$	$\{1, 2, \dots, D\}$
$d$	Jarak antara <i>node</i> dengan <i>centroid</i>
$\overline{X}_i$	Vektor nilai <i>input</i> $\{x_1, x_2, \dots, x_n\}$ .
$\overline{Y}_j$	Vektor nilai <i>centroid</i> $\{y_1, y_2, \dots, y_n\}$ .
$\mu_k$	<i>Centroid</i> baru dari <i>cluster</i> ke- $k$ .
$N_k$	Banyaknya data pada <i>cluster</i> ke- $k$ .
$x_i$	Vektor nilai <i>node</i> ke- $i$ pada <i>cluster</i> ke- $k$ .
$I$	Obyek data ke- $i$
$x_{i,j}$	Sumber makanan awal- $i$ pada dimensi ke- $j$
$l_{jk}$	Nilai bawah dari tiap nilai yang ada di dimensi- $j$ untuk tiap <i>cluster</i>
$u_{jk}$	Nilai atas dari tiap dimensi- $j$ untuk tiap <i>cluster</i>
$E$	Nilai <i>fitness</i>
$\ z_i - C_j\ ^2$	<i>Sum of Squared Error</i> (SSE) data $z_i$ ke <i>centroid</i> $C_j$
$v_{ij}$	Kandidat posisi makanan baru berdasarkan posisi yang lama dalam memori.
$\phi_{ij}$	Angka acak diantara $\{-1, 1\}$
$P$	Nilai Probabilitas
$A$	Rata-rata jarak $i$ terhadap obyek lain pada <i>cluster</i> yang sama.
$B$	Nilai minimum dari rata-rata jarak $i$ terhadap obyek lain pada <i>cluster</i> yang berbeda.
$Si$	Nilai koefisien <i>silhouette</i>

***(Halaman ini sengaja dikosongkan)***



# **BAB I**

## **PENDAHULUAN**

Bab ini akan menjelaskan hal dasar pengerjaan Tugas Akhir yang meliputi: latar belakang, tujuan, manfaat permasalahan, batasan permasalahan, metodologi serta sistematika penulisan. Latar belakang berisi mengenai hal-hal yang melatarbelakangi pemilihan topik Tugas Akhir. Rumusan masalah memuat permasalahan yang diharapkan dapat ditemukan solusinya lewat pengerjaan Tugas Akhir. Batasan masalah berisi batasan-batasan yang melingkupi pembuatan Tugas Akhir. Tujuan berisi hal-hal yang merupakan tujuan pengerjaan Tugas Akhir. Metodologi membahas mengenai metode pengerjaan Tugas Akhir. Sistematika penulisan membahas sistematika penulisan buku Tugas Akhir sebagai laporan dari pengerjaan Tugas Akhir. Penjelasan tentang hal-hal tersebut diharapkan dapat memberikan gambaran umum mengenai permasalahan sehingga penyelesaian masalah dapat dipahami dengan baik.

### **1.1 Latar Belakang**

Pada saat ini terdapat hingga milyaran pengguna dari berbagai negara, yang terdaftar pada bermacam-macam media sosial yang ada, seperti *Facebook* ataupun *Twitter*. *Twitter* sendiri merupakan suatu sumber data sosial yang kaya, yang merupakan titik awal yang amat baik untuk melakukan penggalian data (data mining) sosial karena sifat keterbukaan untuk konsumsi publik, API yang rapi dan terdokumentasi dengan baik, banyaknya *developer tool*, dan merupakan daya tarik yang luas kepada pengguna dari bermacam sudut pandang hidup [1].

Salah satu fenomena yang muncul pada berbagai media sosial tersebut yaitu ketika tahun berganti, ada banyak orang yang mendeklarasikan resolusi akan apa yang mereka inginkan dari diri mereka sendiri untuk dapat tercapai [1]. Pada umumnya, resolusi-resolusi tersebut dapat digeneralisasi menjadi beberapa kelompok

tertentu. Hasil generalisasi tersebut nantinya dapat diolah sebagai suatu sumber informasi yang bermanfaat.

Tugas akhir ini bertujuan untuk mencoba membuat suatu perangkat lunak yang dapat melakukan deteksi dan klasifikasi resolusi tahun baru dari data media sosial *Twitter*. Data yang digunakan adalah *dataset* berisi *tweet* tentang resolusi tahun baru pengguna *Twitter*. Dari sejumlah atribut data yang termasuk dalam *dataset* tersebut, data *kategori resolusi* dan *tweet* menjadi data yang akan diolah untuk mendapatkan informasi yang diinginkan.

Hasil yang ingin diperoleh dari pengerjaan tugas ini adalah pengelompokan dari *dataset* resolusi tahun baru dalam bentuk kelompok-kelompok atau *cluster*. *Cluster* yang terbentuk dapat menunjukkan berbagai hal, antara lain komunitas yang dapat mencakup sejumlah pengguna *Twitter* yang memiliki tingkat kemiripan yang cukup tinggi secara optimal. *K-means* merupakan salah satu algoritma *clustering* yang umum digunakan karena kemudahan dan relatif cepatnya waktu yang dibutuhkan dalam menjalankan pembelajaran [2].

Meskipun demikian, metode *K-means* bukannya tanpa masalah. Khususnya pada tugas akhir ini, pengolahan data *Twitter* dilakukan dengan membentuk fitur yang diperoleh dari kata-kata penting (*keyword*) pada tiap *tweet*. Banyaknya data yang diolah akan mengakibatkan pembengkakan jumlah fitur (dimensi) yang diproses. Tentunya jumlah yang terlalu banyak dapat mengakibatkan berbagai permasalahan ketika pemrosesan algoritma dilakukan, mulai dari kecepatan pemrosesan hingga akurasi dan efisiensi hasil yang diperoleh, yang sering disebut sebagai *Curse of Dimensionality* [2]. Untuk menghindari permasalahan, beberapa solusi dapat diterapkan, diantaranya melakukan perubahan pada algoritma, atau melakukan transformasi pada data ke dalam dimensi yang lebih kecil. Tugas akhir ini akan dibuat dengan menggunakan metode reduksi fitur *Latent Semantic Analysis* (LSA) demi mengatasi permasalahan tersebut.

Permasalahan lain pada *K-means* adalah bahwa metode tersebut sangat bergantung pada penentuan titik awal *centroid* [3]. Titik awal *centroid* biasanya ditentukan secara acak, yang dapat mengakibatkan hasil pengelompokan berbeda-beda setiap kali dijalankan [2]. Untuk mendapatkan titik inisialisasi *centroid* yang lebih optimal, pada tugas akhir ini akan digunakan algoritma *Artificial Bee Colony* (ABC) untuk melengkapi penggunaan *K-Means*. ABC sendiri juga merupakan salah satu algoritma yang umum digunakan, seperti dalam penyelesaian *Traveling Salesman Problem*. Penggunaan metode ABC sebagai media optimasi untuk *K-Means* diharapkan dapat meningkatkan akurasi pemuatan kelompok *cluster*.

## 1.2 Rumusan Masalah

Rincian permasalahan yang dihadapi dalam pembuatan tugas akhir ini adalah sebagai berikut:

1. Bagaimana melakukan pengelompokan resolusi yang diperoleh dari data *tweet* dengan menggunakan gabungan metode *clustering K-means* dan *Artificial Bee Colony* (ABCKM).
2. Bagaimana melakukan evaluasi akurasi untuk mengetahui seberapa tepat hasil pengelompokan yang diperoleh dibandingkan dengan data asli.

## 1.3 Batasan Masalah

Permasalahan yang diajukan dalam pembuatan tugas akhir ini akan diberi batasan untuk membatasi lingkup pengerjaan tugas akhir. Batasan-batasan tersebut adalah sebagai berikut:

1. *Dataset* yang digunakan adalah data resolusi tahun baru dalam bahasa Inggris dari [https://hanna4210.carto.com/tables/new\\_years\\_resolution\\_s\\_dfe/publicasdsa](https://hanna4210.carto.com/tables/new_years_resolution_s_dfe/publicasdsa).

Data yang digunakan adalah data *tweet* dan kategori *tweet* tersebut. Contoh data yang digunakan yaitu sebagai berikut:

- *Tweet* : “#NewYearsResolution :: Read more books, No scrolling FB/checking email b4 breakfast, stay dedicated to PT/yoga to squash my achin' back!”
  - Kategori Resolusi : Health & Fitness
2. *Preprocessing* data menggunakan daftar kata (corpus) yang diperoleh dari *library* untuk Python yaitu *Natural Language Toolkit* (NLTK) <http://www.nltk.org/>.
  3. Metode yang digunakan untuk mengelompokkan hasil data yang telah diolah adalah metode *clustering K-means* dengan optimasi ABC.
  4. Pengelompokan akan dibentuk sesuai jumlah kategori yang terdapat pada *dataset*, yaitu 10 kategori.

## 1.4 Tujuan Tugas Akhir

Tugas Akhir ini memiliki tujuan sebagai berikut:

1. Melakukan implementasi algoritma ABCKM untuk menciptakan kelompok *cluster* dari data resolusi tahun baru.
2. Melakukan pengolahan data resolusi tahun baru untuk diproses ke dalam algoritma ABCKM
3. Melakukan evaluasi algoritma ABCKM dengan cara melakukan implementasi dan uji coba.

## 1.5 Manfaat

Selain untuk menguji efisiensi algoritma ABCKM pada permasalahan pengelompokan resolusi tahun baru, data yang dihasilkan dari tugas akhir ini diharapkan dapat dimanfaatkan untuk keperluan tertentu, misalnya untuk menganalisa

kecenderungan perilaku masyarakat yang akan muncul dalam setahun yang akan datang.

## **1.6 Metodologi**

Pembuatan Tugas Akhir ini terbagi menjadi beberapa tahap, adapun tahapan-tahapan yang dilakukan dalam pengerjaan Tugas Akhir ini adalah sebagai berikut:

### **1.6.1 Penyusunan Proposal Tugas Akhir**

Tahap awal pengerjaan Tugas Akhir adalah menyusun proposal Tugas Akhir. Gagasan yang diajukan pada proposal Tugas Akhir ini adalah menggabungkan algoritma *Artificial Bee Colony* dengan *K-means* dan mengimplementasikannya pada data resolusi tahun baru dari *Twitter*.

### **1.6.2 Studi Literatur**

Tahap berikutnya adalah mencari dan mempelajari informasi dan data, khususnya dalam algoritma ABC dan *K-means*, melalui pengidentifikasian masalah, pencarian studi literatur, perumusan masalah, dan penetapan tujuan penelitian. Dasar informasi yang diperlukan diantaranya mengenai karakteristik algoritma ABC dan *K-means*, algoritma gabungan ABCKM yang akan dikembangkan untuk menyelesaikan permasalahan pembagian data hingga mencapai hasil optimal, dan metode ekstraksi dan pemrosesan data yang diambil dari *Twitter*. Informasi dan literatur didapatkan dari buku, jurnal ilmiah, internet, maupun sumber-sumber informasi lain yang berhubungan.

### **1.6.3 Desain dan Perancangan Sistem**

Setelah memperoleh pemahaman dasar tentang algoritma ABCKM yang akan digunakan, dilakukan pembuatan desain

sistem termasuk didalamnya penysunan struktur algoritma dengan pembuatan *flowchart* dan *pseudocode* yang akan digunakan sebagai acuan untuk tahap pengerjaan berikutnya. Perancangan desain awal system diharapkan dapat membantu menuntun dan memudahkan pengembangan perangkat lunak.

#### 1.6.4 Implementasi Perangkat Lunak

Pada Tahapan ini dilakukan pembangunan dari rancangan aplikasi yang dibuat dengan merealisasikan rancangan yang telah dibuat sebelumnya ke dalam suatu perangkat lunak yang sesuai dengan ruang lingkup Tugas Akhir. Implementasi perangkat lunak dibuat menjadi suatu aplikasi *code* yang ditulis dalam bahasa pemrograman *Python*. Implementasi dilakukan dengan menggunakan perangkat lunak PyCharm sebagai media pengembangan aplikasi dan pelaksanaan *debugging*. Hasil perangkat lunak yang akan diimplementasikan diharapkan mampu membuat solusi optimal dalam pembuatan kelompok *cluster* dari data *Twitter* yang digunakan.

#### 1.6.5 Pengujian dan Evaluasi

Uji coba akan dilakukan pada perangkat lunak yang telah dibuat untuk memperoleh hasil pengelompokan data dan tingkat akurasi dari pengelompokan tersebut. Evaluasi akurasi dilakukan dengan membandingkan *cluster* yang muncul dengan pengelompokan tiap *Tweet* yang terdapat dalam *dataset* pada kolom kategori resolusi. Tahap ini dilakukan juga untuk mengevaluasi jalannya aplikasi, mencari *bug* yang mungkin muncul pada aplikasi, dan memperbaiki aplikasi apabila terdapat kesalahan dalam proses eksekusinya. Uji coba juga akan dilakukan dengan menggunakan perubahan variasi dari variabel tidak tetap yang memengaruhi algoritma. Diharapkan dengan

melakukan evaluasi tersebut maka program dapat disempurnakan dan dapat ditemukan hasil paling optimal.

## 1.7 Sistematika Penulisan Laporan Tugas Akhir

Sistematika penulisan laporan Tugas Akhir adalah sebagai berikut:

1. Bab I. Pendahuluan  
Bab ini berisi penjelasan mengenai latar belakang masalah, tujuan, dan manfaat dari pembuatan Tugas Akhir. Selain itu juga turut disertakan pula rumusan permasalahan, batasan permasalahan, dan sistematika penulisan pada bab ini.
2. Bab II. Tinjauan Pustaka  
Bab ini berisi penjelasan secara detail mengenai dasar-dasar penunjang untuk mendukung pembuatan Tugas Akhir. Dasar ilmu tersebut antara lain algoritma *Artificial Bee Colony* dan *K-means*.
3. Bab III. Desain dan Implementasi  
Bab ini berisi penjelasan mengenai desain metode dan implementasi algoritma yang diusulkan untuk Tugas Akhir ini.
4. Bab IV. Implementasi  
Bab ini berisi tentang penjelasan dan pembahasan mengenai implementasi perangkat lunak yang dibuat.
5. Bab V. Pengujian dan Evaluasi  
Bab ini berisi penjelasan mengenai data hasil percobaan dan pembahasan mengenai hasil percobaan yang telah dilakukan.
6. Bab VI. Kesimpulan dan Saran  
Bab ini berupa hasil penelitian yang menjawab permasalahan atau yang berupa konsep, program, dan rancangan. Selain itu pada bab ini diberikan saran-saran yang berisi hal-hal yang berhubungan dengan penelitian yang telah dilakukan untuk dikembangkan lebih lanjut atau berisi masalah-masalah yang dialami pada proses pengerjaan Tugas Akhir.

7. Daftar Pustaka  
Bab ini berisi daftar pustaka yang dijadikan literatur dalam pengerjaan Tugas Akhir.
8. Lampiran Tugas Akhir  
Dalam lampiran terdapat tabel-tabel data hasil uji coba dan kode sumber program keseluruhan.



## BAB II

### TINJAUAN PUSTAKA

Bab ini membahas tentang teori dasar yang menunjang pengerjaan serta penyusunan Tugas Akhir mengenai penggunaan algoritma *Artificial Bee Colony* yang telah dimodifikasi dengan algoritma *K-means* sehingga menjadi *Artificial Bee Colony K-means*.

#### 2.1 *K-means*

*K-means* merupakan salah satu algoritma *clustering* yang bertujuan membagi data ke dalam beberapa kelompok. *Clustering* sendiri adalah suatu metode pengelompokan yang berbeda dari klasifikasi, dimana pada *clustering* dilakukan pengelompokan data atau objek tanpa mengetahui target kelas yang mungkin akan dihasilkan, atau yang disebut dengan *unsupervised learning*.

Teknik *clustering* dengan *K-means* cukup sederhana [4]. Pada mulanya, pilih titik sejumlah *K* sebagai titik pusat (*centroid*) awal, dimana *K* adalah jumlah *cluster* yang diinginkan user sebagai hasil akhir. Selanjutnya setiap titik yang ada dihubungkan dengan *centroid* dengan jarak terdekat dari dirinya, dan setiap kelompok titik yang terhubung dengan *centroid* merupakan sebuah *cluster*. *Centroid* dari setiap *cluster* tersebut kemudian diperbaharui menjadi titik yang merupakan nilai tengah dari *cluster*. Langkah-langkah tersebut diulangi hingga titik *centroid* stabil tidak mengalami perubahan posisi dari langkah sebelumnya, atau dengan batasan jumlah perulangan yang ditentukan oleh pengguna. Tahapan *K-means* yang digunakan diperjelas dengan pseudocode berikut:

1. Pilih *K* buah titik *centroid* awal (dapat ditentukan sendiri atau acak).
2. Hitung jarak titik (*node*) dengan tiap *centroid*.

3. Kelompokkan titik dengan centroid terdekat. sehingga terbentuk K buah *cluster* dengan titik *centroid* dari setiap *cluster* merupakan titik *centroid* yang telah dipilih sebelumnya
4. Perbaharui nilai titik *centroid* dengan titik tengah baru dari *cluster*.
5. Ulangi langkah 2-5 hingga *centroid* stabil nilainya tidak berubah, atau jika perulangan (*loop*) telah mencapai batasan yang ditentukan.

Untuk perhitungan jarak  $d$  antara *node* dengan *centroid*, digunakan perhitungan vektor yang disesuaikan dengan jumlah dimensi data seperti berikut :

$$d(\overline{X_i}, \overline{Y_j}) = \sqrt{(x_1 - y_1)^2 + \dots + (x_n - y_n)^2} \quad (1)$$

Dimana:

- $\overline{X_i}$  = Vektor nilai *input*  $\{x_1, x_2, \dots, x_n\}$ .
- $\overline{Y_j}$  = Vektor nilai *centroid*  $\{y_1, y_2, \dots, y_n\}$ .
- $n$  = Jumlah dimensi data

Untuk perhitungan perubahan letak *centroid* pada tiap perulangan dapat dilakukan dengan formula:

$$\mu_k = \frac{1}{N_k} \sum_{i=1}^{N_k} x_i \quad (2)$$

Dimana:

- $\mu_k$  = *Centroid* baru dari *cluster* ke-k.
- $N_k$  = Banyaknya data pada *cluster* ke-k.
- $x_i$  = Vektor nilai *node* ke-i pada *cluster* ke-k.

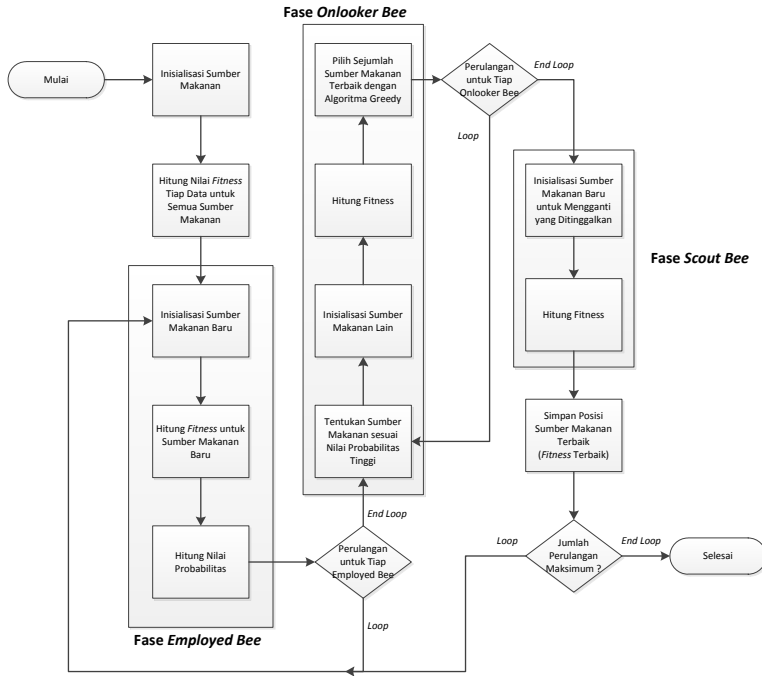
Memilih titik pusat suatu *cluster* merupakan tugas yang rumit karena diatur maupun diacak, apabila nilai inisialisasi yang dilakukan kurang baik maka pengelompokan data bisa kurang optimal [3].

Selain penentuan titik pusat, permasalahan lain yang umum ditemui pada *K-means* berhubungan dengan jumlah dimensi data yang diolah. Untuk menghindari hal tersebut, pada tugas akhir ini digunakan metode reduksi dimensi *Latent Semantic Analysis* (LSA).

## 2.2 *Artificial Bee Colony*

*Artificial Bee Colony* (ABC) adalah sebuah algoritma yang digunakan untuk optimasi yang dibuat dengan didasari oleh perilaku pencarian makanan oleh kawanan lebah madu. ABC digunakan untuk memecahkan permasalahan optimasi multidimensional dan multimodal [5].

Pada pemodelan algoritma ABC, kawanan lebah digolongkan menjadi tiga kelompok, yaitu *employed bee*, *onlooker bee*, dan *scout bee*. Jumlah *employed bee* ditentukan sama dengan jumlah sumber makanan di sekitar sarang lebah. Setiap *employed bee* memiliki tujuan sumber makanan masing-masing. *Employed bee* kemudian pergi ke sumber makanan tujuannya dan kembali ke sarang untuk memberikan informasi tentang banyak makanan yang tersisa pada sumber makanan dan daerah sekitarnya. *Onlooker bee* menerima informasi tersebut dan memilih satu sumber makanan untuk dikunjungi, kemudian mengevaluasi kualitas sumber makanan tersebut. *Scout bee* menentukan mana saja sumber makanan yang perlu ditinggalkan, dan menggantinya dengan yang baru. Langkah-langkah tersebut diulangi hingga mendapatkan sumber makanan yang terbaik.



**Gambar 2.1 Diagram Alir Artificial Bee Colony**

Pada ABC, posisi sumber makanan  $SN$  mewakili solusi yang mungkin untuk permasalahan optimasi dan jumlah madu pada sumber makanan merujuk pada kualitas (*fitness level*) untuk solusi tersebut. Jumlah kawanan lebah disamakan dengan jumlah sumber makanan yang ditentukan. Pada proses inisiasi sumber makanan  $x_{i,j}$  digunakan variabel acak yang ditentukan menggunakan distribusi normal antara 0 hingga 1 sesuai dengan formula :

$$x_{i,j} = l_{jk} + rand(0,1)(u_{jk} - l_{jk}) \quad (3)$$

dimana  $i$  mewakili nilai dari 1 hingga jumlah  $SN$ ,  $j$  adalah nilai dari 1 hingga jumlah dimensi  $D$ ,  $k$  adalah nilai dari 1 hingga jumlah  $cluster$ ,  $u_{jk}$  dan  $l_{jk}$  merupakan batas atas dan bawah dari parameter dimensi yang dipilih untuk tiap  $cluster$  yang dipilih.

Perhitungan nilai *fitness* dalam pengerjaan aplikasi selanjutnya disusun dengan mempertimbangkan bahwa algoritma ABC akan digunakan bersama dengan *K-means*. Berdasarkan hal tersebut, nilai *fitness* ditentukan sebagai nilai yang disebut *distorsi*, yang merupakan nilai *sum of squared error* (SSE) yang didefinisikan seperti berikut :

$$E = \sum_{j=1}^K \sum_{z_i \in c_j} \| z_i - C_j \|^2 \quad (4)$$

dimana  $(z_i - C_j)$  merepresentasikan jarak tiap *node* yang bersesuaian dengan titik *centroid* dari *cluster* masing-masing. Nilai distorsi  $E$  dihitung untuk tiap sumber makanan, dengan semakin kecil nilainya menunjukkan bahwa sumber makanan tersebut memiliki *fitness* yang lebih baik.

Inisialisasi sumber makanan baru dilakukan kembali pada fase *employed bee*, namun dibuat berdasarkan sumber makanan yang telah ditemukan sebelumnya. *Employed bee* akan mencari secara acak sumber makanan baru  $v_{ij}$  yang berada di sekitar sumber makanan lama  $z_{ij}$  dengan rincian perhitungan seperti berikut :

$$v_{ij} = z_{ij} + \varphi(z_{ij} - z_{kj}) \quad (5)$$

dimana  $i$  dan  $k$  adalah dua sumber makanan yang berbeda,  $j$  merupakan dimensi, dan  $\varphi$  adalah nilai random yang berdistribusi *uniform* antara  $[-1, 1]$ .

Nilai probabilitas yang akan digunakan pada fase *onlooker bee* diperoleh dari nilai *fitness* yang diberikan oleh *employed bee*. *Onlooker bee* memilih sumber makanan berdasarkan dari nilai probabilitas tersebut. Ada kemungkinan lebih dari satu *onlooker*

*bee* yang memilih sumber makanan yang sama. Nilai probabilitas  $P$  itu sendiri dihitung dengan cara seperti berikut:

$$P_i = 1 - \frac{E_i}{\sum_{j=1}^{SN} (1 - E_j)} \quad (6)$$

Nilai yang diambil dikurangkan dari satu untuk membuat sumber makanan terbaik (nilai *distorsi* kecil) memiliki probabilitas lebih tinggi untuk dapat dipilih oleh *onlooker bee*.

Semua sumber makanan yang diproses memiliki variabel *counter* yang berfungsi untuk mencatat jumlah sumber makanan tersebut dipilih oleh semua tipe *bee*. Sumber makanan yang memiliki jumlah *counter* melebihi batas (*limit*) yang ditentukan akan digunakan pada fase *scout bee* untuk mencari sumber makanan baru, kemudian *counter* sumber makanan yang lama dikembalikan menjadi nol lagi. Pencarian sumber makanan menggunakan rumus yang sama dengan proses pada fase *employed bee* (3).

Ketiga fase proses *employed*, *onlooker*, dan *scout bee* akan terus diulang hingga kriteria terminasi perulangan dipenuhi dan sumber makanan terbaik yang diperoleh akan dipilih sebagai solusi terakhir.

### 2.3 Term Frequency – Inverse Domain Frequency

Informasi tertentu yang diperoleh dari suatu kumpulan data (*dataset*) berjumlah besar tidak bisa diolah begitu saja. Diperlukan suatu proses yang disebut *data mining*, yaitu proses yang bertujuan untuk mendapatkan informasi dari suatu *dataset* dan mengubahnya kedalam suatu bentuk yang dapat dipahami. Sesuai dengan tujuan Tugas Akhir ini yaitu mengolah data dari *Twitter* menjadi data yang berkelompok, proses yang dilakukan secara keseluruhan termasuk ke dalam *data mining*.

Proses *data mining* yang digunakan dalam Tugas Akhir ini berhubungan dengan pengolahan data berupa teks. Sebelum pengelompokan data dapat dilakukan, data mentah perlu diolah

untuk menunjukkan suatu nilai dalam bentuk vektor numerik. Proses ekstraksi fitur dari *dataset* diawali dengan pencarian kata kunci (keyword) yang diperoleh dari seluruh *tweet* pada *dataset*. Kata kunci akan digunakan sebagai *token*, kemudian dihitung frekuensi kemunculan *token* pada *dataset*. Dari hasil tersebut, fitur diperoleh dengan menggunakan metode perhitungan *Term Frequency-Inverse Domain Frequency* (TF-IDF).

TF-IDF sendiri adalah suatu statistik numerik yang menunjukkan seberapa penting suatu kata dari kumpulan dokumen (*corpus*). Nilai TF-IDF meningkat sebanding dengan banyaknya kemunculan kata pada satu dokumen, namun menurun apabila kata tersebut juga banyak muncul pada dokumen lain dalam *corpus*. Struktur perhitungan nilai TF-IDF adalah sebagai berikut:

$$tfidf(t, d, D) = tf(t, d) * idf(t, D) \quad (7)$$

dimana  $tf$  merupakan jumlah kemunculan kata  $t$  pada dokumen  $d$ , dan perhitungan  $idf$  dilakukan seperti berikut:

$$idf(t, D) = \log \frac{N}{|\{d \in D : t \in D\}|} \quad (8)$$

dimana  $N$  merupakan jumlah dokumen pada corpus, dan  $|\{d \in D : t \in D\}|$  menunjukkan banyaknya jumlah dokumen yang mengandung kata  $t$  didalamnya. Pada Tugas Akhir ini, dokumen merupakan substitusi untuk tiap *tweet* pada *dataset*.

## 2.4 Latent Semantic Analysis

Proses seleksi fitur perlu dilakukan dalam data mining untuk memilah variabel data yang tidak diperlukan dan melakukan penyederhanaan pemodelan agar memudahkan penafsiran data untuk proses selanjutnya [6]. Salah satu metode ekstraksi fitur yang termasuk efektif untuk diterapkan adalah metode reduksi dimensi, dimana dalam ruang lingkup Tugas

Akhir ini digunakan untuk mencegah terjadinya Curse of Dimensionality, yaitu permasalahan yang muncul pada proses pengelompokan akibat terlalu banyak fitur yang diproses.

*Latent Semantic Analysis* (LSA) merupakan metode pemrosesan untuk menganalisis hubungan antara sekumpulan dokumen dengan kata yang termuat didalamnya dengan membentuk suatu konsep yang berdasarkan kemunculan kelompok kata pada dokumen. LSA mengasumsikan bahwa kata yang memiliki arti serupa akan muncul bersamaan pada suatu bagian dokumen. Kumpulan kelompok kata tersebut dihitung nilainya dan dilakukan pengurangan jumlah baris kata dengan mempertahankan struktur kemiripan antar dimensi, menggunakan teknik yang disebut *Singular Value Decomposition* (SVD).

## 2.5 *Silhouette*

Nilai *silhouette* digunakan untuk menganalisis performa algoritma pembentukan kelompok-kelompok (*cluster*) data, khususnya untuk *Kmeans*. *Silhouette* merupakan suatu metode untuk melakukan validasi konsistensi dalam *cluster*. Nilai *silhouette* mengindikasikan seberapa mirip suatu objek dalam *cluster* dengan *clusternya* sendiri dibandingkan dengan *cluster* lainnya. Nilai *si* tersebut dapat diinterpretasikan seperti berikut:

$$si = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \quad (9)$$

dimana  $a(i)$  menunjukkan besaran perhitungan ketidakmiripan data ke- $i$  dengan *clusternya* sendiri, dan  $b(i)$  adalah rata-rata terendah ketidakmiripan data  $i$  dengan *clusternya* sendiri.

**Tabel 2.1 Interpretasi Nilai *Silhouette* [7]**

RANGE NILAI SILHOUETTE	INTERPRETASI



0.71-1.0	Struktur <i>cluster</i> yang kuat ditemukan
0.51-0.70	Struktur yang cukup baik ditemukan
0.26-0.50	Struktur lemah dan kemungkinan tidak benar. Cobalah gunakan metode analisis data tambahan,
$\leq 0.25$	Tidak ada struktur substansial yang ditemukan

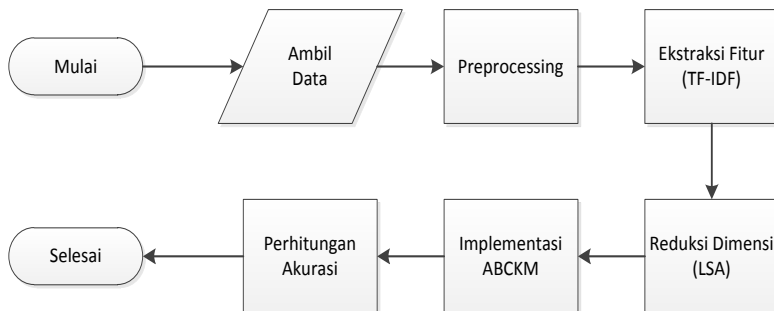
Nilai *silhouette* selalu berada diantara  $-1$  dan  $1$ , dengan semakin besar nilainya, maka kualitas *cluster* yang diperoleh semakin baik. Tabel 2.1 menunjukkan interpretasi nilai *silhouette*.

*(Halaman ini sengaja dikosongkan)*

### BAB III

## DESAIN DAN PERANCANGAN PERANGKAT LUNAK

Pada bab ini akan dibahas perancangan sistem aplikasi perangkat lunak yang dikembangkan untuk Tugas Akhir ini. Aplikasi dibuat menggunakan algoritma yang berdasarkan metode *Artificial Bee Colony* dan *K-means* (ABCKM) yang digunakan untuk menetapkan pengelompokan resolusi tahun baru dari *dataset* yang diambil dari *Twitter*. Secara garis besar, rancangan keseluruhan sistem yang digunakan dapat dilihat pada diagram alir pada gambar 3.1.



**Gambar 3.1 Diagram Alir Keseluruhan Sistem**

Proses kerja perangkat lunak dimulai dengan pengambilan data dari *Twitter*, yang kemudian diolah pada tahap *preprocessing* untuk menghasilkan nilai yang kemudian dapat dibagi menjadi kelompok *cluster*. Setelah *preprocessing*, dilakukan analisis performa dengan menggunakan nilai *silhouette* yang diambil dengan penggunaan *K-means* secara umum untuk mendapatkan besaran jumlah dimensi yang dinilai optimal, yaitu nilai *silhouette*

yang melebihi 0.5. Jumlah dimensi digunakan kembali untuk pengolahan *dataset* yang kemudian diproses pada metode ABCKM. Keseluruhan proses dilakukan berulang menggunakan jumlah sumber makanan dan iterasi ABCKM yang berbeda sebagai gambaran hasil keseluruhan penerapan algoritma pada Tugas Akhir ini.

### 3.1 Ekstraksi Data dan Preprocessing

*Dataset* resolusi tahun baru yang diambil dari *Twitter* merupakan data berbentuk teks yang ditulis menggunakan Bahasa Inggris. Setiap satuan data (*record*) memiliki 15 macam atribut, namun yang digunakan untuk proses pengelompokan pada Tugas Akhir ini hanya 2 atribut, yaitu '*Resolution Category*' yang merupakan klasifikasi asli dari data, dan atribut '*Text*' yang merupakan isi dari tiap *tweet*.

Data *tweet* kemudian diolah melalui proses *preprocessing* untuk memberikan nilai yang dapat digunakan dalam metode ABCKM. Proses *preprocessing* tersebut mencakup beberapa tahapan yang utamanya ditujukan agar data yang digunakan kemudian merupakan data yang baik dan terbebas dari data yang kurang relevan (*noise*). Berikut adalah tahapan yang digunakan dalam Tugas Akhir ini :

- *Parsing*

Proses pemisahan (*parsing*) dilakukan untuk membagi *tweet* menjadi satuan per kata, agar dapat diolah kemudian. Parsing dilakukan dengan memotong *tweet* pada tiap karakter spasi yang terdapat didalamnya. Setiap kata yang terbentuk dikelompokkan sesuai dengan *tweet* asalnya.

- Menghapus data yang tidak relevan

Data yang tidak digunakan dalam pengerjaan tugas ini akan dihapus. Diantara data yang dihapus adalah, data yang

mengandung tautan (*link*), mengandung tagar (*hashtag*), juga data yang mengandung karakter yang bukan alfabet.

- Menghapus *stopword* (dalam bahasa Inggris)

*Stopword* adalah kata umum (*common words*) yang biasanya muncul dalam jumlah besar dan dianggap tidak memiliki makna. Kehadiran *stopword* dalam proses data mining biasanya akan mengakibatkan munculnya *noise*, sehingga perlu untuk dihilangkan. Karena *dataset* yang digunakan menggunakan bahasa Inggris, tentunya *stopword* yang digunakan merupakan *stopword* dalam bahasa Inggris, diantaranya yang umum ditemukan adalah *function words*, seperti *the*, *is*, *at*, *which*, *on*; dan masih banyak lagi. Daftar *stopword* yang digunakan pada tugas ini menggunakan daftar kata (*corpus*) pada *library* untuk python, *Natural Language Toolkit* (NLTK). Kata dalam *tweet* yang termasuk dalam *corpus* akan dihapus.

- *Stemming* (dalam bahasa Inggris)

*Stemming* adalah suatu proses untuk mereduksi kata turunan kembali pada kata dasarnya. Contoh proses *stemming* untuk kata berbahasa Inggris seperti mendeteksi sekumpulan kata '*stems*', '*stemmer*', '*stemming*', '*stemmed*' semuanya sebagai kata dengan basis '*stem*' sebagai kata dasarnya. Pada tugas ini, proses *stemming* dilakukan dengan menggunakan fungsi **PorterStemmer** yang termasuk dalam library NLTK.

### 3.2 Analisis Performa dengan Silhouette

Data yang telah melewati tahap *preprocessing* kemudian diolah untuk mendapatkan nilai vektor numerik. Pengolahan data dilakukan dengan ekstraksi fitur yang diawali dengan proses pencarian kata kunci (*keyword*) yang kemudian dilakukan proses *tokenizing*, yaitu pemberian nilai bobot dari tiap *keyword*. Frekuensi kemunculan *token* dihitung untuk kemudian dilakukan

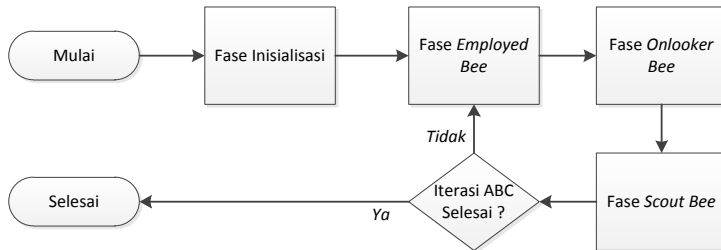
ekstraksi dengan metode *Term Frequency-Inverse Domain Frequency* (TF-IDF).

Reduksi dimensi kemudian dilakukan untuk mencegah terjadinya *Curse of Dimensionality*, yaitu permasalahan yang muncul pada metode pengelompokan akibat terlalu banyak fitur yang diproses. Untuk menghindari permasalahan tersebut, pada tugas akhir ini akan digunakan metode reduksi dimensi *Latent Semantic Analysis* (LSA). Untuk memperkirakan efisiensi reduksi dimensi dilakukan uji coba dengan membentuk kelompok *K-means*, kemudian dihitung nilai *silhouette* untuk sejumlah besaran dimensi yang diuji. Besaran dimensi yang memiliki nilai *silhouette* yang melebihi 0.5 dianggap cukup baik untuk menjadi pilihan dalam pemrosesan menggunakan metode ABCKM.

### 3.3 Aplikasi Metode ABCKM

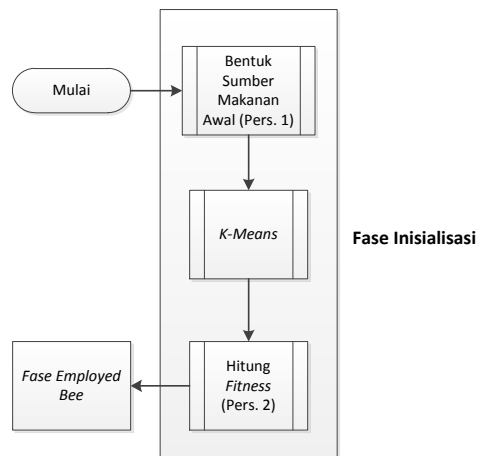
Pembuatan program pada pengerjaan Tugas Akhir ini berpusat pada penggunaan gabungan metode *Artificial Bee Colony* dengan *K-means* (ABCKM). *K-means* digunakan untuk membentuk *cluster* dengan menggunakan metode ABC sebagai optimasi dalam pencarian *centroid*. Penggabungan metode sebagai ABCKM menjadikan ABC sebagai kerangka struktur keseluruhan proses, dengan menyelipkan *K-means* pada setiap fase lebah yaitu *employed bee*, *onlooker bee*, dan *scout bee*. Hasil pembentukan *cluster K-means* pada tiap tahap termasuk pada tahap inisiasi akan digunakan pada tiap fase sesudahnya.

Hasil dari ABCKM adalah sekumpulan sumber makanan yang akan dihitung nilai kelayakan (*fitness*), dengan sumber makanan yang memiliki nilai *fitness* terbaik akan menjadi solusi terakhir. *K-means* dilakukan untuk terakhir kali pada sumber makanan terbaik untuk mendapatkan *cluster* terakhir, yang kemudian akan dihitung besaran akurasi yang didapatkan dari pengelompokan yang terbentuk.



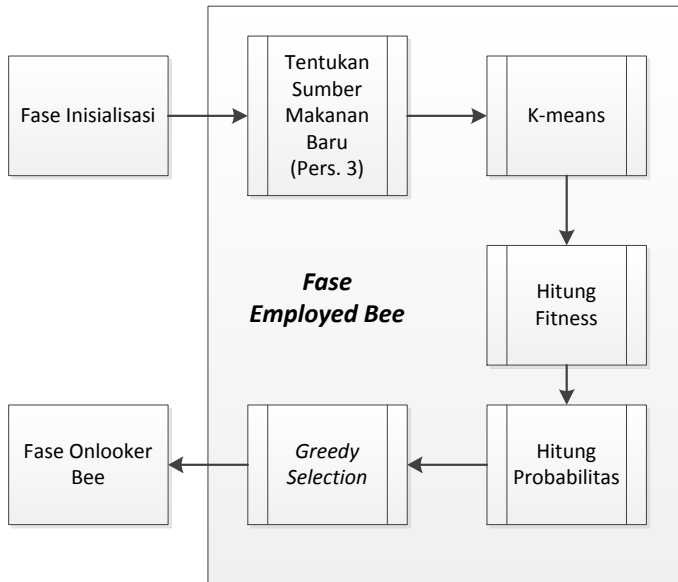
**Gambar 3.2 Diagram Alir ABCKM Sederhana**

ABCKM dimulai dengan fase inisialisasi menggunakan data yang telah melalui pemrosesan bentuk menjadi numerik. Sumber makanan awal dibentuk secara acak sesuai dengan persamaan 3. *K-means* dilakukan dengan menggunakan sumber makanan tersebut sebagai acuan *centroid*, kemudian nilai *fitness* dari *cluster* yang terbentuk dihitung menggunakan persamaan 4. Tiap sumber makanan yang terbentuk akan dicatat untuk perhitungan jumlah kemunculan (*counter*).



**Gambar 3.3 Diagram Alir Fase Inisialisasi**

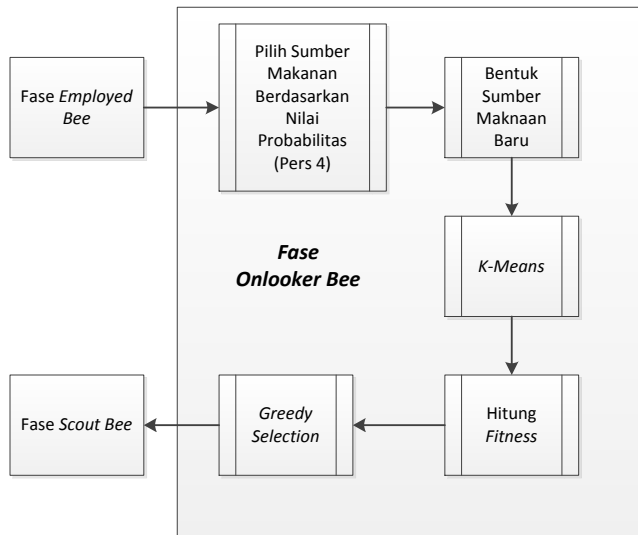
Sumber makanan yang telah terbentuk kemudian digunakan pada fase *employed bee*, yang akan mencari sumber baru yang berdekatan dengan sumber hasil inisialisasi menggunakan persamaan 5. Sumber makanan yang baru kemudian digunakan untuk *K-means*, dan setelahnya dihitung nilai *fitness* untuk masing-masing sumber. Disini sumber makanan yang terbentuk juga dicatat dan apabila terdapat sumber dengan kemunculan yang berulang maka nilai *counter* sumber tersebut akan ditambah. Setelah nilai *fitness* diperoleh, perhitungan nilai probabilitas digunakan sesuai persamaan 6 yang akan digunakan selanjutnya pada fase *onlooker bee*. Sumber makanan yang terbentuk pada fase ini dibandingkan dengan sumber pada fase inisialisasi kemudian dipilih dengan menggunakan algoritma *greedy* sejumlah batas sumber makanan yang ditentukan.



**Gambar 3.4 Diagram Alir Fase *Employed Bee***



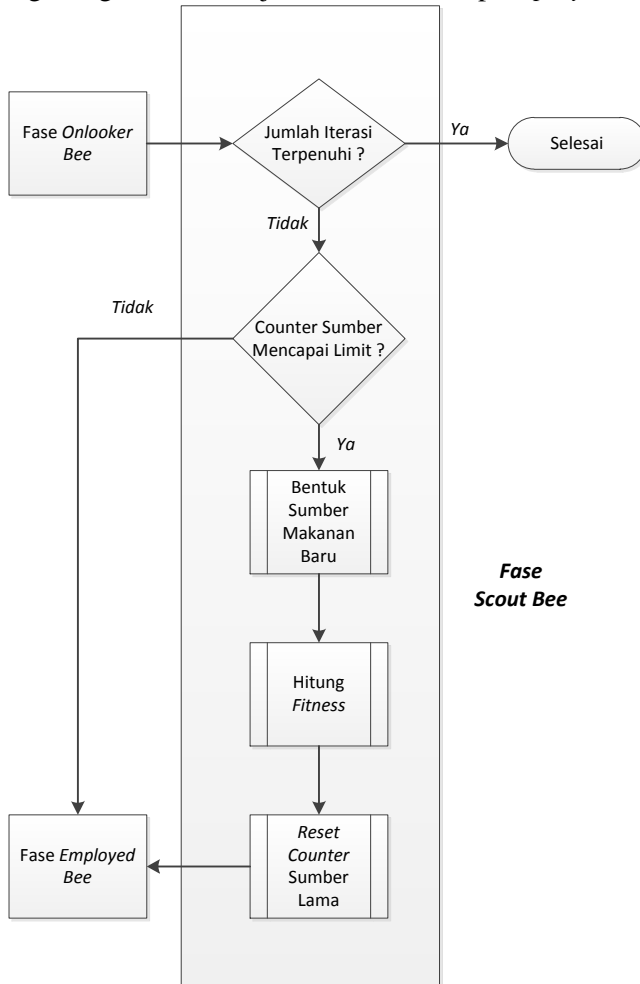
Pada fase *onlooker bee*, sumber makanan dari fase *employed bee* dipilih secara acak, dengan kemungkinan sumber yang memiliki nilai *fitness* lebih baik mendapatkan kemungkinan lebih besar untuk terpilih. Dari sumber makanan yang terpilih dicari sumber makanan baru menggunakan kembali persamaan 5. *K-means* dan perhitungan *fitness* kembali dilakukan untuk sumber baru, kemudian dilakukan perhitungan *greedy* untuk mengambil sumber makanan terbaik yang telah terbentuk antara fase *employed bee* dan *onlooker bee*.



**Gambar 3.5 Diagram Alir Fase Onlooker Bee**

Fase *scout bee* dijalankan apabila jumlah iterasi maksimal proses belum dipenuhi. Pada fase ini, sumber makanan yang telah mencapai batas maksimal kemunculan akan diganti menjadi sumber baru menggunakan persamaan 5. Nilai *counter* dari Sumber makanan yang tergantikan tersebut akan diulang kembali

mulai dari nol. Sumber makanan baru akan dibawa melewati perulangan algoritma menuju kembali ke tahap *employed bee*.



**Gambar 3.6 Diagram Alir Fase Scout Bee**

Ketiga fase *employed*, *onlooker*, dan *scout bee* akan terus diulang hingga jumlah iterasi terpenuhi. Sumber makanan terbaik yang didapatkan setelah itu akan dipilih menjadi solusi terakhir. *K-means* dilakukan kembali menggunakan sumber tersebut untuk membentuk kelompok *cluster* sebagai hasil uji coba untuk perhitungan akurasi.

*(Halaman ini sengaja dikosongkan)*

## BAB IV IMPLEMENTASI

Bab ini menjelaskan mengenai implementasi Tugas Akhir yang meliputi algoritma dan kode sumber yang digunakan untuk melakukan proses pengelompokan pada *dataset* resolusi tahun baru dari *Twitter* dengan menggunakan algoritma ABCKM.

### 4.1. Lingkungan Implementasi

Implementasi perangkat lunak dilakukan dengan menggunakan perangkat keras komputer dengan prosesor AMD A8-6410 APU (4 CPUs), ~1.60GHz yang mempunyai memori sebesar 4.0 GB. Sistem operasi yang digunakan adalah *Windows* 8.1 Pro 64-bit. Implementasi dilakukan dengan bantuan program *JetBrains PyCharm* sebagai perangkat pengembangan (IDE) menggunakan bahasa pemrograman *Python*.

### 4.2 Implementasi

Implementasi sistem perangkat lunak dilakukan sesuai dengan proses yang telah dijelaskan pada bab sebelumnya. Program utama ABCKM akan dijelaskan dan dijabarkan lebih lanjut pada subbab berikutnya.

#### 4.2.1 Implementasi Tahap Preprocessing

Pada bagian awal program dilakukan deklarasi *library* dan variabel utama yang akan digunakan selanjutnya. Beberapa variabel tersebut yaitu: jumlah *cluster* yang akan dibentuk, batas iterasi ABC, jumlah sumber makanan, dan batas maksimal perulangan suatu sumber makanan. Kemudian tahap preprocessing dimulai dengan proses ekstraksi data, melakukan *parsing*, penghapusan data tidak relevan, penghapusan *stopword*,

dan *stemming*. Secara keseluruhan, implementasi preprocessing ditunjukkan oleh Kode Sumber 4.1.

```

36  ### Data Retrieval + Stopword Removal + Stemming ###
37  ifile = codecs.open('New-years-resolutions-DFE.csv',
38  "rb")
39  reader2 = csv.reader(x.replace('\0', ' ') for x in ifile)
40
41  tweet = []
42  rownum = 0
43
44  count = 0
45  for row in reader2:
46      if (count != 0):
47          tweet.append([row[4], row[6]])
48          rownum+=1
49          count+=1
50
51  preprocessed = []
52
53  for x in range(0, rownum):
54      words = re.split(' +', tweet[x][1])
55      merged = ""
56      for inWord in words:
57          inWord = re.sub(r'^http?:\V.*[\r\n]*', '', inWord)
58          inWord = re.sub(r'^\x00-\x7F'+', ' ', inWord)
59
60      if inWord.startswith('#'):
61          inWord = None
62      elif inWord.startswith('@'):

```

62	inWord = None
63	if inWord != None:
64	inWord = re.sub(r'^a-zA-Z', '', inWord)
	if inWord.lower() in (cachedStopWords): inWord =
65	None
	if inWord != None: inWord =
66	PorterStemmer().stem_word(inWord)
67	if inWord != None: merged = merged + " " + inWord
68	preprocessed.append(merged)
69	##### End Pengolahan Data #####

#### Kode Sumber 4.1 Implementasi Preprocessing

Proses dimulai pada baris ke-37 dimana data diambil dari *dataset* yang masuk pada file 'New-years-resolutions-DFE.csv', dimana pada baris berikutnya, data kosong (*null*) yang mungkin terdapat pada *dataset* akan diganti dengan karakter kosong agar dapat dimasukkan dalam pemrosesan selanjutnya. Pada baris ke-44 hingga 48 dilakukan pengambilan data hanya pada kolom yang akan digunakan, yaitu kolom 'tweet' dan 'resolution category'. Selanjutnya dimulai dari baris ke-52 dilakukan proses pengambilan kata kunci (*keyword*) dari setiap *tweet*. Pada baris ke-56 hingga 62 dilakukan penghapusan data yang tidak relevan. Pada baris ke-56 data yang dihapus adalah data yang merupakan tautan, pada baris berikutnya adalah data dengan karakter diluar ASCII, baris ke-59 dan 60 menghapus data yang berupa tagar (*hashtag*), baris ke-61 dan 62 menghapus data yang berupa *mention* pada *tweet*. Pada baris ke-64 hingga 67 data dibagi untuk tiap kata, kemudian dilakukan pengecekan *stopword* dan *stemming* menggunakan bantuan *library*, dan pada akhirnya kata-kata tersebut digabungkan lagi untuk tiap baris data *tweet*.

#### 4.2.2 Implementasi Reduksi Dimensi LSA

Tahap selanjutnya yaitu melakukan reduksi dimensi menggunakan algoritma Latent Semantic analysis yang implementasinya dilakukan dengan bantuan library. Implementasi proses tersebut ditunjukkan pada Kode Sumber 4.2.

```

71 ##### LSA Dimension Reduction #####
72 from sklearn.feature_extraction.text import
73 CountVectorizer
74 vect = CountVectorizer(min_df=2, max_df=setMaxDf)
75 vect.fit(preprocessed)
76
77 from sklearn.feature_extraction.text import
78 TfidfTransformer
79 ft_extracted = vect.fit_transform(preprocessed)
80 tf_transformer = TfidfTransformer().fit(ft_extracted)
81 tf_idf = tf_transformer.transform(ft_extracted)
82
83 from sklearn.decomposition import TruncatedSVD
84 from sklearn.pipeline import make_pipeline
85 from sklearn.preprocessing import Normalizer
86
87 svd = TruncatedSVD(n_components = setDimensionLSA)
88 normalizer = Normalizer(copy=False)
89 lsa = make_pipeline(svd, normalizer)
90 lsa_fit = lsa.fit_transform(tf_idf)
91
92 explained_variance = svd.explained_variance_ratio_.sum()

```



```

93  print lsa_fit
94  print("Explained variance of the SVD step:
      {}%".format(int(explained_variance * 100)))
95  ##### End LSA Dimension Reduction #####

```

#### Kode Sumber 4.2 Implementasi Reduksi Dimensi LSA

### 4.2.3 Implementasi Inisialisasi Sumber Makanan

Inisialisasi sumber makanan diawali dengan pengurutan data untuk tiap fitur. Implementasi tahap ini ditunjukkan pada Kode Sumber 4.3. Pada tahap ini data yang telah diurutkan dibagi rata sejumlah banyaknya *cluster* yang akan dibentuk (baris ke-111). Nilai terendah pada setiap kelompok yang terbentuk akan dijumlahkan dengan bilangan acak berdistribusi normal antara 0 dan 1 (baris ke-114 dan 115).

```

105  ##### Start Fase Inisialisasi #####
106  sn = []
107  n1 = 0
108
109  while n1 < snNum:
110      for y in range(0, colnum):
111          dataSorted = sorted(data, key=lambda data: data[y])
112          centro = []
113          for x in range(0, setClusterNumKmeans):
114              1) randNum = random.randint(0, nodePerClusterNum-
115                  randNum]
116                  centro.append(nodeCenter)
117                  sn.append(centro)
118                  n1 += 1

```

```

119
120 snInit = np.array(sn)
121 ##### End Fase Inisialisasi #####

```

#### **Kode Sumber 4. 3 Implementasi Inisialisasi Sumber Makanan**

#### **4.2.4 Implementasi Optimasi Inisialisasi Sumber Makanan**

Pada tahap ini dilakukan beberapa proses sekaligus, yaitu perhitungan *K-means* untuk sumber makanan yang telah diinisialisasi, perhitungan nilai *fitness* untuk hasil *K-means*, kemudian penyimpanan sumber makanan yang terbentuk pada memori yang nantinya digunakan untuk mengetahui sumber makanan yang terbaik dan jumlah perulangan kemunculan setiap sumber makanan.

```

125 #####Start Optimasi Inisialissasi (K-Means) #####
126 kmSN = []
127 resInitSN = []
128 for y in range(0, snNum):
129     km = KMeans(n_clusters=setClusterNumKmeans,
130               max_iter=setMaxIterKmeans, n_init=1, init=snInit[y])
131     res = km.fit(data)
132     kmSN.append(res.cluster_centers_)
133     y+=1
134 ##### Hitung Fitness #####
135 initDistance = km.fit_transform(data)
136 initLabel = km.labels_
137 counter = 0
138 initDistanceSumSquare = 0
139 for abc in initDistance:

```

```

139         initDistanceSumSquare = initDistanceSumSquare +
abc[initLabel[counter]]
140         counter+=1
141         initFitness = initDistanceSumSquare
142         ##### End Hitung Fitness #####
143         ##### Counter SN #####
144         tempArr = []
145         tempArr.append(res.cluster_centers_)
146         tempArr.append(initFitness)
147         resInitSN.append(tempArr)
148         if (y == 0):
149             tempArr2 = []
150             tempArr2.append(res.cluster_centers_)
151             tempArr2.append(initFitness)
152             tempArr2.append(1)
153             snCounter.append(tempArr2)
154         else:
155             assigned = 0
156             countNum = 0
157             for snA in snCounter:
158                 countNum += 1
159             for klz in range (0, countNum):
160                 if(np.all(res.cluster_centers_ == snCounter[klz][0])):
161                     assigned = 1
162                     snCounter[klz][2] += 1
163             if (assigned == 0):
164                 tempArr2 = []
165                 tempArr2.append(res.cluster_centers_)
166                 tempArr2.append(initFitness)
167                 tempArr2.append(1)

```

168	snCounter.append(tempArr2)
169	##### End Counter SN #####

#### Kode Sumber 4.4 Implementasi Optimasi dengan *K-means*

Disini *K-means* dilakukan menggunakan bantuan *library* ‘*sklearn*’. Pada baris ke-130 fungsi dari *library* digunakan dengan masukan jumlah *cluster* yang akan dibentuk yaitu 10, jumlah iterasi maksimal *K-means* yang dilakukan apabila *cluster* yang dihasilkan tidak kunjung berbentuk optimal, dan *centroid* awal yang ditetapkan menggunakan hasil dari tahap inisialisasi sumber makanan sebelumnya.

*Cluster* yang dihasilkan proses *K-means* ditetapkan menjadi sumber makanan baru. Kemudian dihitung nilai *fitness* untuk masing-masing sumber makanan tersebut yang diimplementasikan sesuai Kode Sumber 4.4 pada baris ke-133 hingga 142. *initDistance* pada baris ke-136 merupakan jarak tiap *node* data ke *centroid* yang terbentuk. *initLabel* pada baris berikutnya menunjukkan *cluster* dimana setiap data ditempatkan pada akhir *K-means*. Jarak pada *initDistance* merupakan nilai *sum of squared error* (SSE) yang merupakan nilai distorsi. Nilai distorsi tiap data tersebut kemudian dijumlahkan pada baris ke-141 sehingga membentuk nilai *fitness* dari sumber makanan.

Tahap yang terakhir adalah penyimpanan sumber makanan yang berupa *centroid* untuk masing-masing *cluster* dan nilai *fitness* dari sumber makanan tersebut. Baris ke-148 melakukan penyimpanan sumber makanan jika sumber tersebut merupakan yang pertama diproses. Untuk semua sumber berikutnya (baris ke-154 hingga 168), dilakukan pengecekan apabila sumber makanan yang sedang diproses sama dengan sumber makanan yang telah disimpan, ditunjukkan pada baris ke-160. Apabila sumber telah tersimpan maka jumlah kemunculan akan ditambahkan satu. Di sisi lain, sumber makanan akan disimpan apabila sebelumnya belum ditemukan.

#### 4.2.5 Implementasi *ABCKM* pada Fase *Employed Bee*

Pada tahap berikutnya, implementasi ABC dilakukan, dimulai dengan fase *employed bee*. Implementasi fase *employed bee* ditunjukkan pada Kode Sumber 4.5. Pada fase ini dilakukan pencarian sumber makanan baru menggunakan sumber makanan yang telah diinisialisasi sebelumnya sebagai dasar, yang diimplementasikan pada baris ke-184 hingga 187. Sumber baru dibentuk dengan penjumlahan (baris ke-186) tiap fitur pada tiap *centroid* dari masing-masing sumber makanan dengan nilai acak berdistribusi uniform antara -1 dan 1 (baris ke-185).

179	<code>employedSN = []</code>
180	<code>for y in resInitSN:</code>
181	<code>tempSN = []</code>
182	<code>for centroNow in y[0]:</code>
183	<code>tempCentro = []</code>
184	<code>for cc in centroNow:</code>
185	<code>randNum2 = random.uniform(-1, 1)</code>
186	<code>var = float(cc) + randNum2</code>
187	<code>tempCentro.append(var)</code>
188	<code>tempSN.append(tempCentro)</code>
189	<code>employedSN.append(tempSN)</code>
190	
191	<code>empSN = np.array(employedSN)</code>

**Kode Sumber 4.5 Implementasi *ABCKM*  
pada Fase *Employed Bee***

Selanjutnya dihitung nilai *fitness* tiap sumber makanan yang terbentuk, dan penyimpanan sumber makanan yang dihasilkan, seperti pada tahap sebelumnya. Implementasi kedua tahap tersebut serupa dengan tahap sebelumnya, hanya mengubah

variabel masukan menjadi sumber makanan baru yang dihasilkan fase *employed bee*.

Kemudian sumber makanan terbaik yang dihasilkan antara kedua fase inisialisasi dan *employed bee* dipilih menggunakan *greedy selection*. Algoritma *greedy* diimplementasikan dengan mengurutkan sumber makanan sesuai nilai *fitness* yang tertinggi hingga terkecil, kemudian memilih sumber makanan tersebuturut dari yang terbaik hingga sejumlah banyaknya sumber yang akan disimpan. Sumber makanan yang tidak terpilih tetap disimpan dalam *counter* sumber makanan, namun tidak akan digunakan pada fase-fase berikutnya.

232	##### Greedy Selection #####
233	tempBest = []
234	for ax in resInitSN:
235	tempBest.append(ax)
236	for ay in resEmpSN:
237	tempBest.append(ay)
238	newTemp = sorted(tempBest, key=lambda sra: sra[1])
239	snBest = []
240	for kll in range (0, snNum):
241	snBest.append(newTemp[kll])
242	viewSN = np.array(snBest)
243	##### End Greedy Selection #####

**Kode Sumber 4.6 Implementasi *Greedy Selection*  
pada Fase *Employed Bee***

#### 4.2.6 Implementasi *ABCKM* pada Fase *Onlooker Bee*

Sumber makanan dari fase *employed bee* akan dipilih untuk digunakan pada fase *onlooker bee* berdasarkan nilai probabilitasnya. Semakin baik nilai *fitness* suatu sumber makanan, maka semakin besar kemungkinan sumber makanan tersebut terpilih pada fase *onlooker bee*. Implementasi pencarian sumber baru pada fase *onlooker bee* ditunjukkan pada Kode Sumber 4.7.

```

259 onlookerSN = []
260
261 ##### Hitung Probabilitas #####
262 probabilitas = []
263 count = 0
264 sumFitness = 0
265 for temp in snBest:
266     sumFitness = sumFitness + (1-temp[1])
267     count+=1
268
269 for q in snBest:
270     probabilitas.append((1-q[1])/sumFitness)
271 ### Probabilitas End ###
272
273 resOnSN = []
274
275 for az in range(0, snNum):
276     randNum3 = random.random()
277     cumulative = 0
278     selected = 0
279     for k in range(0, count):
280         if (cumulative < randNum3):

```

```

281         selected = k
282         cumulative = cumulative + probabilitas[k]
283
284         selectedSN = snBest[selected][0]
285
286         tempSN = []
287         for sc in selectedSN:
288             tempCentro = []
289             for xc in sc:
290                 randNum4 = random.uniform(-1, 1)
291                 var = float(xc) + randNum4
292                 tempCentro.append(var)
293             tempSN.append(tempCentro)
294         #-----#
295         onSN = np.array(tempSN)

```

**Kode Sumber 4.7 Implementasi *ABCKM* pada  
Fase *Onlooker Bee***

Perhitungan nilai probabilitas dilakukan pada baris ke-262 hingga 270. Nilai probabilitas yang menunjukkan kualitas sumber makanan akan semakin besar apabila nilai *fitness* sumber tersebut kecil. Selanjutnya suatu sumber dipilih secara acak berdasarkan nilai probabilitas yang telah dibentuk (baris ke-275 hingga 284). Dari sumber makanan yang telah terpilih, dilakukan pencarian sumber makanan baru yang serupa dengan pencarian sumber pada fase *employed bee* (baris ke-286 hingga 295).

Setelah sumber baru terbentuk, kembali diterapkan optimasi dengan *K-means*, perhitungan *fitness*, penyimpanan *counter* sumber makanan, dan *greedy selection*. Sumber makanan terbaik yang dihasilkan algoritma *greedy* digunakan pada fase selanjutnya.



#### 4.2.7 Implementasi *ABCKM* pada Fase *Scout Bee*

Pada fase *scout bee*, jumlah kemunculan (counter) tiap sumber makanan diperiksa. Sumber makanan dengan nilai counter yang telah mencapai batas (limit) akan digunakan sebagai dasar pencarian sumber baru. Nilai counter sumber makanan lama akan dikembalikan menjadi nol lagi (reset).

359	resScSN = []
360	count = 0
361	for akb in snCounter:
362	count += 1
363	for kzl in range (0, count):
364	if (snCounter[kzl][2] > maxTrialCounter):
365	tempSN = []
366	tempSN.append(snCounter[kzl][0])
367	tempSN.append(snCounter[kzl][1])
368	scSN = np.array(tempSN)
369	snCounter[kzl][2] = 0
370	
371	scKM = KMeans(n_clusters=setClusterNumKmeans, max_iter=setMaxIterKmeans, n_init=1, init=scSN[0])
372	resSc = scKM.fit(data)

**Kode Sumber 4.8 Implementasi *ABCKM* pada Fase *Scout Bee***

Apabila ada, sumber makanan yang dihasilkan pada fase ini akan digunakan melalui perulangan tahapan algoritma ABC kembali ke fase *employed bee*. Namun apabila jumlah batas perulangan ABC telah terpenuhi maka sumber terbaik yang telah dihasilkan sejauh itu akan digunakan sebagai hasil akhir algoritma *ABCKM* dan perulangan tidak dilakukan kembali.

#### 4.2.8 Implementasi Klasifikasi Dataset Menggunakan Sumber Makanan Terbaik

Tahap ini merupakan tahap terakhir dari keseluruhan algoritma. Klasifikasi dilakukan dengan pembentukan *cluster* menggunakan *K-means* dengan sumber makanan terbaik digunakan sebagai *centroid* awal. Menggunakan kategori resolusi yang didapatkan dari *dataset* awal, dihitung jumlah data dari tiap *cluster* terbentuk yang seharusnya masuk dalam klasifikasi kategori awal. Impelementasi tersebut ditunjukkan pada Kode Sumber 4.9.

```

422 lastBest = sorted(snBest, key=lambda fit: fit[1])
423 print ('Sumber Makanan (Solusi) Terbaik :')
424 print np.array(lastBest[0][0])
425 print ('Fitness = %f' %lastBest[0][1])
426
427 endKM = KMeans(n_clusters=setClusterNumKmeans,
428               max_iter=setMaxIterKmeans, n_init=1, init=lastBest[0][0])
428 endRes = endKM.fit(data)
429 endResLabel = endRes.labels_
430
431 #####Klasifikasi Hasil Akhir #####
432
433 category = []
434 category_num = 0
435 count = 0
436
437 for res in tweet:
438     assigned = 0
439     for xyz in range(0, category_num):

```

```

440         if res[0] == category[xyz][0]:
441             category[xyz][(endResLabel[count-1]) + 1 ]+=1
442             assigned = 1
443         if assigned == 0:
444             add_category = []
445             add_category.append(res[0])
446             for lup in range (0, setClusterNumKmeans):
447                 if lup == endResLabel[count-1]:
448                     add_category.append(1)
449                     category.append(add_category)
450                     category_num+=1
451             count +=1
452
453     print ('Hasil akhir klasifikasi')
454     for result in category:
455         print result

```

**Kode Sumber 4.9 Implementasi Klasifikasi *Dataset***

*(Halaman ini sengaja dikosongkan)*

## **BAB V**

### **HASIL UJI COBA DAN EVALUASI**

Pada bab ini akan dijelaskan rangkaian pengujian dan evaluasi untuk uji coba metode ABCKM dalam menyelesaikan permasalahan pengelompokan data resolusi tahun baru dari *Twitter*. Hal yang akan dibahas meliputi lingkungan uji coba, skenario uji coba, hasil uji coba dan evaluasi uji coba yang dilakukan.

#### **5.1 Lingkungan Pengujian**

Uji coba dan evaluasi Tugas Akhir ini menggunakan perangkat keras komputer dengan prosesor AMD A8-6410 APU (4 CPUs), ~1.60GHz yang mempunyai memori sebesar 4.0 GB. Sistem operasi yang digunakan adalah *Windows 8.1 Pro 64-bit*. Implementasi dilakukan dengan bantuan program *JetBrains PyCharm* sebagai perangkat pengembangan (IDE) menggunakan bahasa pemrograman *Python*, serta Microsoft Office Excel 2013 untuk pengolahan beberapa tabel yang terdapat pada Tugas Akhir ini.

#### **5.2 Skenario Uji Coba**

Data masukan yang digunakan pada Tugas Akhir ini adalah data resolusi tahun baru yang diperoleh dari *Twitter*. Data diolah dalam bentuk file dengan format *Comma Separated Values* (CSV). File yang digunakan dinamakan 'New-years-resolutions-DFE.csv', berisikan 15 kolom atribut, namun untuk tugas akhir ini atribut yang relevan hanya 2 kolom yaitu 'Resolution Category' yang digunakan sebagai kelas acuan (*benchmark*) dan 'text' yang merupakan data teks dari *tweet*.

Uji coba awal dilakukan untuk penetapan variabel kunci pada program, diantaranya jumlah *cluster* yang dibentuk, dimensi yang digunakan, sumber makanan, dan iterasi ABC.

Jumlah *cluster* ditetapkan sesuai dengan jumlah kelas pada *benchmark* yaitu 10, sedangkan jumlah sumber makanan dan iterasi ABC akan digunakan sebagai variabel yang divariasi dengan batasan antara 20-100 untuk uji coba selanjutnya. Percobaan untuk mengetahui jumlah dimensi dari reduksi dimensi LSA dilakukan menggunakan implementasi *K-means* dengan 10 *cluster* yang hasilnya akan dianalisis menggunakan nilai *silhouette*.

**Tabel 5.1 Uji Perbandingan Besaran Dimensi LSA**

Fitur LSA	Cluster	Nilai Silhouette pada Percobaan ke-			Nilai Silhouette Rata-rata
		1	2	3	
2	10	0.583	0.567	0.570	0.573
3		0.453	0.416	0.454	0.441
4		0.376	0.366	0.362	0.368
5		0.320	0.351	0.290	0.321
6		0.323	0.339	0.328	0.330
7		0.336	0.306	0.307	0.316
8		0.312	0.316	0.309	0.312
9		0.319	0.325	0.283	0.309
10		0.314	0.272	0.334	0.307

Hasil percobaan tersebut menunjukkan bahwa nilai *silhouette* terbaik dibentuk ketika menggunakan reduksi dimensi fitur menjadi dua dimensi. Rata-rata nilai *silhouette* yang dihasilkan adalah 0.573 yang menunjukkan bahwa *cluster* yang terbentuk cukup baik. Namun, untuk percobaan selanjutnya penggunaan hanya dua dimensi dirasa kurang dalam mewakili variasi data yang digunakan. Berdasarkan pertimbangan tersebut, jumlah dimensi yang akan digunakan selanjutnya adalah lima, meskipun nilai *silhouette* yang

terbentuk kalah baik dibandingkan ketika menggunakan jumlah dimensi dua.

### 5.3 Uji Kinerja

Pada tahap ini dilakukan pengujian kinerja program untuk permasalahan pembagian data. Uji kinerja ini dilakukan dengan cara mencari akurasi pengelompokan dan nilai *fitness* sumber makanan yang dihasilkan setelah menggunakan algoritma ABCKM dengan menggunakan variasi untuk parameter jumlah sumber makanan dan iterasi ABC. Untuk variabel dimensi dan jumlah *cluster* ditentukan sama untuk semua percobaan, yaitu jumlah dimensi adalah lima dan jumlah *cluster* adalah 10.

Nilai *fitness* suatu sumber makanan sebagaimana yang telah dijelaskan pada bab 2, merupakan nilai total dari jarak sebuah *node* (data) menuju *cluster* yang terdekat darinya, yang dihitung dengan *Sum of Squared Error* (SSE). Semakin kecil nilai *fitness* yang dihasilkan, maka semakin baik *cluster* yang terbentuk. Berikutnya akan dibahas hasil pengujian program dengan variasi variabel sumber makanan dan iterasi ABC yang digunakan, juga hasil dari penerapan algoritma *K-means* saja yang dioptimasi.

Tabel 5.2 menunjukkan perbandingan nilai *fitness* menggunakan variasi variabel dari implementasi algoritma ABCKM. Nilai *fitness* diperoleh dari rata-rata nilai yang dihasilkan dalam percobaan dengan masing-masing besaran iterasi dan sumber makanan (SN). Uji coba untuk setiap variasi variabel dilakukan sebanyak minimal tiga kali. Demikian juga untuk uji coba hanya menggunakan hasil *K-means* sebanyak tiga kali percobaan.

**Tabel 5.2 Nilai *Fitness* Hasil Uji ABCKM**

Fitness		Sumber Makanan						AVG
		KM	20	30	50	75	100	
	KM	1241.24						
Iterasi	25		1223.57	1224.31	1227.25	1222.22	1221.96	<b>1223.86</b>
	50		1230.54	1231.08	1225.52	1224.37	1221.77	<b>1226.66</b>
	75		1224.46	1225.85	1223.73	1220.98	1223.01	<b>1223.61</b>
	100		1230.02	1223.41	1222.92	1223.47	1219.93	<b>1223.95</b>
	AVG		<b>1227.15</b>	<b>1226.16</b>	<b>1224.85</b>	<b>1222.76</b>	<b>1221.67</b>	

Hasil dari percobaan menunjukkan bahwa secara keseluruhan, nilai *fitness* yang dihasilkan dengan menggunakan hanya *K-means*, yaitu **1241.24**, lebih tinggi daripada menggunakan ABCKM, yaitu antara nilai terendah diperoleh dari pengujian dengan 100 SN dan 100 iterasi (**1219.93**) dan nilai tertinggi diperoleh dari pengujian menggunakan 50 SN dan 50 iterasi (**1231.08**), menunjukkan bahwa *cluster* yang terbentuk pada ABC lebih baik.

Selanjutnya perhitungan akurasi dilakukan untuk mengetahui seberapa akurat pengelompokan yang dihasilkan melalui penggunaan ABCKM. Ketepatan pengelompokan data akan dihitung dengan berdasar pada kelas *benchmark* yang diperoleh dari *dataset* yang berjumlah 10 kelas. Perhitungan yang digunakan berdasar pada *confusion matrix*, yang merupakan perbandingan data aktual dengan data hasil dari pengujian. *Confusion matrix* menunjukkan banyaknya data uji yang masuk dalam kelas bersesuaian dengan data asal (*True Positive*/TP), data yang seharusnya tidak termasuk dalam suatu kelas (*False Positive*/FP), data yang seharusnya termasuk dalam suatu kelas (*False Negative*/FN), dan data yang memang tidak termasuk dalam suatu kelas (*True Negative*/TN). Metode penilaian yang digunakan yaitu akurasi dengan metode *Rand Measure*, nilai prediksi positif atau *Precision*, dan nilai sensitifitas atau *Recall*. Ketiga metode perhitungan menggunakan *confusion matrix* sebagai dasar.



**Tabel 5.3 Nilai Akurasi *Rand Measure*  
Hasil Uji ABCKM**

Rand Measure			Sumber Makanan					AVG
	KM		20	30	50	75	100	
	KM	83.07%						
Iterasi	25		83.20%	83.10%	83.17%	83.14%	83.14%	<b>83.15%</b>
	50		82.99%	83.06%	83.13%	83.11%	83.14%	<b>83.09%</b>
	75		83.19%	83.11%	83.09%	83.10%	83.17%	<b>83.13%</b>
	100		83.13%	83.14%	83.11%	83.09%	83.08%	<b>83.11%</b>
	AVG		<b>83.13%</b>	<b>83.10%</b>	<b>83.13%</b>	<b>83.11%</b>	<b>83.13%</b>	

Tabel 5.3 menunjukkan akurasi yang dihasilkan setelah uji coba dilakukan. Nilai akurasi *Rand Measure* dihasilkan menggunakan perhitungan berikut.

$$Akurasi = \frac{TP+TN}{TP+FP+FN+TN} \quad (8)$$

Akurasi umumnya dihasilkan lebih baik ketika jumlah iterasi yang digunakan adalah 25 (**83.15%**). Rata-rata akurasi terburuk justru didapatkan ketika uji coba dilakukan menggunakan 100 SN dengan 100 kali iterasi (**83.08%**). Sedangkan akurasi yang dihasilkan dengan hanya menggunakan *K-means* (**83.07%**) masih lebih rendah dari penggunaan metode ABCKM dengan berapapun jumlah SN dan iterasi.

Tabel 5.4 dan 5.5 masing-masing menunjukkan perhitungan nilai *precision* dan *recall* dari uji coba yang dilakukan. Persamaan 11 dan 12 menunjukkan metode perhitungan kedua nilai tersebut.

$$Precision = \frac{TP}{TP+FP} \quad (11)$$

$$Recall = \frac{TP}{TP+FN} \quad (12)$$

Nilai *precision* dihasilkan dari penggunaan *K-means* saja ternyata lebih unggul dari sebagian besar penggunaan ABCKM. Nilai rata-rata *precision* dengan *K-means* adalah **11.19%**, hanya kalah dari penggunaan ABCKM dengan iterasi sebanyak 25 kali dengan SN sebesar 30 dan 50 yang keduanya bernilai **11.38%**, dan iterasi 75 dengan SN 30 dan 50 yang masing-masing bernilai **11.43%** dan **11.3%**.

**Tabel 5.4 Nilai *Precision* Hasil Uji ABCKM**

Precision		Sumber Makanan						AVG
		KM	20	30	50	75	100	
	KM	11.19%						
Iterasi	25		10.90%	11.38%	11.38%	11.02%	11.18%	<b>11.17%</b>
	50		11.17%	11.04%	11.06%	11.12%	11.08%	<b>11.09%</b>
	75		11.17%	11.43%	11.30%	10.94%	11.04%	<b>11.18%</b>
	100		11.06%	11.23%	11.03%	11.15%	11.10%	<b>11.11%</b>
	AVG		<b>11.08%</b>	<b>11.27%</b>	<b>11.19%</b>	<b>11.06%</b>	<b>11.10%</b>	

**Tabel 5.5 Nilai *Recall* Hasil Uji ABCKM**

Recall		Sumber Makanan						AVG
		KM	20	30	50	75	100	
	KM	11.98%						
Iterasi	25		11.78%	12.40%	12.47%	12.28%	12.44%	<b>12.27%</b>
	50		12.22%	11.62%	12.31%	12.39%	12.31%	<b>12.17%</b>
	75		12.33%	12.46%	12.36%	12.22%	12.16%	<b>12.31%</b>
	100		12.35%	12.36%	12.27%	12.29%	12.30%	<b>12.32%</b>
	AVG		<b>12.17%</b>	<b>12.21%</b>	<b>12.35%</b>	<b>12.29%</b>	<b>12.30%</b>	

Nilai recall terbaik yang muncul pada uji coba adalah ketika digunakan 100 SN dengan 25 kali iterasi (**12.44%**). Disini nilai recall algoritma *K-means* saja (**11.98%**) masih kalah dibandingkan sebagian besar penggunaan ABCKM. Meskipun demikian masih terdapat nilai recall lebih rendah dari *K-means*, yaitu pada penggunaan 20 SN dengan 25 kali iterasi (**11.78%**) dan 30 SN dengan 50 kali iterasi (**11.62%**).

Uji coba kemudian dilakukan dengan melakukan seleksi data yang diambil dari *dataset*. Percobaan ditujukan untuk menganalisa performa algoritma ketika jumlah data antar kelas yang digunakan seimbang, dimana pada *dataset* asal terdapat ketimpangan pada jumlah data antar kelas, khususnya antara kelas *Personal Growth* yang memiliki 1781 data, dibandingkan dengan kelas *Education/Training*, *Time Management/Organization*, dan *Philanthropic* yang memiliki jumlah data dibawah 100.

Data yang digunakan untuk percobaan ini yaitu data dari kelas *Personal Growth*, *Health & Fitness*, dan *Humor* yang dari masing-masing kelas diambil 100 data. Percobaan dilakukan dengan menggunakan jumlah variabel sumber makanan (SN)=20, iterasi ABC=25, dimensi LSA=5, dan cluster=3.

**Tabel 5.6 Hasil Uji ABCKM dengan Seleksi Data**

	Uji Coba ke-			Rata-rata
	1	2	3	
<b>Fitness</b>	164.4526	162.3561	166.7451	164.5179
<b>Precision</b>	34.83%	33.33%	33.83%	33.99%
<b>Recall</b>	37.00%	36.67%	36.33%	36.67%
<b>Akurasi</b>	56.00%	54.78%	55.44%	55.41%

Dari tiga kali percobaan, nilai rata-rata akurasi yang diperoleh adalah sekitar 55%, lebih kecil dibandingkan hasil pengujian dengan keseluruhan *dataset* dengan variabel sumber makanan dan iterasi yang sama, yang memiliki nilai akurasi sekitar 83%. Meskipun demikian, nilai *precision* dan *recall* yang muncul pada percobaan ini lebih baik dibandingkan dengan penggunaan keseluruhan *dataset*, yaitu nilai *precision* 34.83% dibanding 10.9%, dan nilai *recall* 37% dibanding 11.78%.

***(Halaman ini sengaja dikosongkan)***

## BAB VI

### KESIMPULAN DAN SARAN

Isi dari bab terakhir ini meliputi kesimpulan mengenai pengujian dan evaluasi yang telah dilakukan. Disamping itu, terdapat masukan untuk penyempurnaan ke depan.

#### 6.1 Kesimpulan

Kesimpulan yang diperoleh berdasarkan pengujian dan evaluasi yang telah dilakukan adalah sebagai berikut:

1. Metode ABCKM mampu membentuk *cluster* yang baik, ditunjukkan dari nilai *fitness* yang dihasilkan (terendah= **1219.93**, tertinggi=**1231.08**) semuanya lebih baik dari nilai *fitness* menggunakan *K-means* (**1241.24**).
2. Begitu pula dalam hal akurasi, dimana penggunaan ABCKM seluruhnya menunjukkan hasil (**83.08%-83.2%**) lebih baik daripada penggunaan hanya *K-means* (**83.07%**).
3. *Cluster* dengan bentuk yang baik tidak selalu menghasilkan akurasi pengelompokan yang lebih baik. Hal ini ditunjukkan dari pengujian dengan menggunakan 100 SN dengan 100 kali iterasi, dimana nilai *fitness* yang dihasilkan merupakan yang terbaik (**1219.93**), namun nilai akurasi yang dihasilkan merupakan nilai terendah (**83.08%**) dari seluruh pengujian ABCKM.
4. Nilai *precision* yang dihasilkan menggunakan *K-means* (**11.19%**) secara umum lebih baik dari ABCKM (rata-rata=**11.14%**), namun untuk nilai *recall* yang terjadi adalah sebaliknya dimana ABCKM (rata-rata=**12.27%**) secara umum lebih baik daripada *K-means* (**11.98%**) .
5. Nilai *precision* dan *recall* yang dihasilkan rendah, berkisar antara 11-12%. Hal ini kemungkinan disebabkan karena *dataset* yang digunakan sebenarnya

lebih cocok digunakan untuk permasalahan *klasifikasi* yang merupakan metode pembelajaran dimana kelas awal data telah diketahui (*supervised learning*) dan bukan *clustering* pada algoritma *K-means* dan ABCKM yang merupakan metode *unsupervised learning*.

## 6.2 Saran

Terdapat beberapa masukan untuk memperoleh hasil yang bisa jadi lebih baik dari yang diperoleh melalui pengejaan Tugas Akhir ini, yaitu:

1. Melakukan *preprocessing* data yang lebih baik agar hasil yang diperoleh lebih akurat.
2. Perlu dilakukan percobaan menggunakan dimensi yang paling optimal, karena uji coba yang dilakukan disini menggunakan lima dimensi fitur, walaupun dimensi paling optimal menurut pengujian adalah dua.
3. Penambahan variasi jumlah sumber makanan dan iterasi ABC, disamping menambah jumlah uji coba keseluruhan yang dilakukan.
4. Penggunaan metode perhitungan *fitness* selain menggunakan *Sum of Squared Error* (SSE) mungkin dapat menjadikan hasil akhir lebih baik.
5. Menerapkan Metode ABCKM untuk permasalahan *unsupervised learning*, dimana akurasi yang mungkin dihasilkan bisa jadi lebih baik secara keseluruhan, termasuk untuk nilai *precision* dan *recall*.

## DAFTAR PUSTAKA

1. **Rusell, Matthew A. 2014.** *Mining the Social Web 2nd Edition: Data Mining Facebook, Twitter, LinkedIn, Google+, Github, and More.* s.l. : O'Reilly Media, Inc, 2014.
2. **X. Wu, V. Kumar.** *The Top Ten Algorithms in Data Mining.* 2009 : Chapman and Hall.
3. **Armano, G., & Farmani, M. R. 2014.** *Clustering Analysis with Combination of Artificial Bee Colony Algoritihm and k-Means Technique.* s.l. : International Journal of Computer Theory and Engineering, 2014.
4. **P.-N. Tan, M. Steinbach, and V. Kumar. 2005.** *Introduction to Data Mining.* Boston, MA, USA : Addison-Wesley Longman Publishing Co., Inc., 2005.
5. **Karaboga, Dervis. 2005.** *An Idea Based On Honey Bee Swarm for Numerical Optimization.* s.l. : Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
6. **James, Gareth, et al. 2013.** *An Introduction to Statistical Learning.* s.l. : Springer, 2013.
7. **UNESCO.** Partitioning Around Medoids (PAM). *UNESCO | Building Peace in the Minds of Men and Women.* [Online]  
[http://www.unesco.org/webworld/idams/advguide/Chapt7\\_1\\_1.htm](http://www.unesco.org/webworld/idams/advguide/Chapt7_1_1.htm).

*(Halaman ini sengaja dikosongkan)*



## LAMPIRAN A

### SAMPEL HASIL AKHIR UJI COBA

**Tabel Lampiran A 1 Sumber Makanan Terbaik Uji Coba dengan sumber makanan (SN)=20, Iterasi=25, Percobaan ke-1**

Info Percobaan		Cluster	Dimensi ke-				
			1	2	3	4	5
SN	20	1	0.229922	0.145742	0.77361	0.213878	-0.46941
Iterasi ABC	25	2	0.333266	0.133778	0.543569	-0.09186	0.709718
Dimensi	5	3	0.450129	0.203061	0.644928	0.231169	0.48622
		4	0.443539	0.225318	0.784582	0.109886	0.207694
No. Test	2	5	0.72363	0.071705	0.385949	0.374858	-0.24262
Waktu		6	0.240565	0.784743	-0.00293	0.021867	0.08774
		7	0.882879	0.023964	0.174381	0.157254	0.298235
Mulai	15:22	8	0.765314	-0.03393	0.043243	-0.53406	-0.12334
Selesai	15:25	9	0.848676	-0.04232	-0.1961	0.415216	0.042589
		10	0.969857	-0.04394	-0.09064	-0.0158	-0.07027
			Fitness = 1223.597966				

**Tabel Lampiran A 2 Klasifikasi Uji Coba dengan SN=20, Iterasi=25, Percobaan ke-1**

Kelas	Cluster										
Benchmark	4	2	1	3	9	8	7	6	10	5	Total
PG	374	256	245	121	206	185	123	119	108	44	1781
Hu	178	146	134	78	126	116	72	62	66	32	1010
H&F	178	116	130	76	90	74	47	55	50	24	840
R&L	78	75	64	48	48	42	46	27	23	16	467
F/F/R	66	50	56	24	40	28	17	25	24	21	351
Fin	26	21	24	19	21	19	15	16	8	7	176
Car	20	15	21	15	14	11	13	9	8	0	126
E/T	21	9	16	2	9	8	6	8	4	6	89
TM/O	12	17	13	9	7	7	8	3	7	4	87
Phi	13	10	16	9	5	3	9	13	3	3	84
Total	966	715	719	401	566	493	356	337	301	157	

Keterangan untuk kelas *benchmark* :

PG = Personal Growth

H&F = Health & Fitness

F/F/R = Family / Friends / Relationships

Car = Career

TM/O = Time Management / Organization

Hu = Humor

R&L = Recreation & Leisure

Fin = Finance

E/T = Education / Training

Phi = Philanthropic

**Tabel Lampiran A 3 *Confusion Matrix* Uji Coba dengan  
SN=20, Iterasi=25, Percobaan ke-1**

Kelas Benchmark	TP	FP	FN	TN	Total
'Personal Growth'	374	592	1407	2638	5011
'Humor'	146	569	864	3432	5011
'Health & Fitness'	130	589	710	3582	5011
'Recreation & Leisure'	48	353	419	4191	5011
'Family/Friends/Relationships'	40	526	311	4134	5011
'Finance'	19	474	157	4361	5011
'Career'	13	343	113	4542	5011
'Education/Training'	8	329	81	4593	5011
'Time Management/Organization'	7	294	80	4630	5011
'Philanthropic'	3	154	81	4773	5011

**Tabel Lampiran A 4 Perhitungan Akurasi Uji Coba  
dengan SN=20, Iterasi=25, Percobaan ke-1**

Kelas Benchmark	Precision	Recall	Rand Measure
'Personal Growth'	0.387163561	0.209994385	0.601077629
'Humor'	0.204195804	0.144554455	0.714029136
'Health & Fitness'	0.180806676	0.154761905	0.740770305
'Recreation & Leisure'	0.119700748	0.102783726	0.845938934
'Family/Friends/Relationships'	0.070671378	0.113960114	0.832967472
'Finance'	0.038539554	0.107954545	0.874077031
'Career'	0.036516854	0.103174603	0.90900002
'Education/Training'	0.023738872	0.08988764	0.918180004
'Time Management/Organization'	0.023255814	0.08045977	0.925364199
'Philanthropic'	0.01910828	0.035714286	0.953103173
<b>Rata-rata</b>	<b>0.110369754</b>	<b>0.114324543</b>	<b>0.831450808</b>

**Tabel Lampiran A 5 Sumber Makanan Terbaik Uji Coba dengan SN=20, Iterasi=25, Percobaan ke-2**

Info Percobaan		Cluster	Dimensi ke-				
			1	2	3	4	5
SN	20	1	0.335751	0.134944	0.544046	-0.09094	0.70583
Iterasi ABC	25	2	0.227612	0.144876	0.77398	0.215672	-0.47191
Dimensi	5	3	0.448109	0.20581	0.654209	0.230935	0.474253
		4	0.849611	-0.04369	-0.19929	0.413914	0.040062
No. Test	2	5	0.883256	0.021648	0.175923	0.164976	0.289134
		6	0.96839	-0.04153	-0.08877	-0.0074	-0.07483
Waktu		7	0.451737	0.220869	0.785801	0.102209	0.183284
Mulai	15:27	8	0.272285	0.899636	-0.00521	0.03393	0.094686
Selesai	15:31	9	0.548626	0.054932	0.298769	0.307367	-0.19496
		10	0.765754	-0.03212	0.041108	-0.53365	-0.1247
Fitness = 1223.164156							

**Tabel Lampiran A 6 Klasifikasi Uji Coba dengan SN=20, Iterasi=25, Percobaan ke-2**

Kelas	Cluster										
Benchmark	4	2	1	3	9	8	7	6	10	5	Total
PG	383	260	236	205	184	122	125	56	109	101	1781
Hu	184	152	127	122	115	77	77	35	66	55	1010
H&F	189	116	119	89	76	75	48	33	50	45	840
R&L	84	76	60	48	41	46	44	15	27	26	467
F/F/R	69	50	56	40	28	23	17	19	26	23	351
Fin	27	22	22	21	19	19	15	8	8	15	176
Car	20	18	18	13	11	15	12	2	9	8	126
E/T	22	9	15	9	8	2	6	10	4	4	89
TM/O	12	18	12	7	7	9	8	4	7	3	87
Phi	13	10	16	5	3	9	9	4	3	12	84
Total	1003	731	681	559	492	397	361	186	309	292	

**Tabel Lampiran A 7 Confusion Matrix Uji Coba dengan SN=20, Iterasi=25, Percobaan ke-2**

Kelas Benchmark	TP	FP	FN	TN	Total
'Personal Growth'	383	620	1398	2610	5011
'Humor'	152	579	858	3422	5011
'Health & Fitness'	119	562	721	3609	5011
'Recreation & Leisure'	48	511	419	4033	5011
'Family/Friends/Relationships'	28	464	323	4196	5011
'Finance'	19	378	157	4457	5011
'Career'	12	349	114	4536	5011
'Education/Training'	10	176	79	4746	5011
'Time Management/Organization'	7	302	80	4622	5011
'Philanthropic'	12	280	72	4647	5011

**Tabel Lampiran A 8 Perhitungan Akurasi Uji Coba  
dengan SN=20, Iterasi=25, Percobaan ke-1**

Kelas Benchmark	Precision	Recall	Rand Measure
'Personal Growth'	0.381854437	0.215047726	0.597285971
'Humor'	0.207934337	0.15049505	0.713230892
'Health & Fitness'	0.174743025	0.141666667	0.743963281
'Recreation & Leisure'	0.085867621	0.102783726	0.814408302
'Family/Friends/Relationships'	0.056910569	0.07977208	0.84294552
'Finance'	0.047858942	0.107954545	0.893234883
'Career'	0.033240997	0.095238095	0.907603273
'Education/Training'	0.053763441	0.112359551	0.949111954
'Time Management/Organization'	0.022653722	0.08045977	0.923767711
'Philanthropic'	0.04109589	0.142857143	0.92975454
<b>Rata-rata</b>	<b>0.110592298</b>	<b>0.122863435</b>	<b>0.831530633</b>

**Tabel Lampiran A 9 Sumber Makanan Terbaik Uji Coba  
dengan SN=50, Iterasi=50, Percobaan ke-1**

Info Percobaan		Cluster	Dimensi ke-				
			1	2	3	4	5
SN	50	1	0.444844	0.209199	0.65895	0.224893	0.471364
Iterasi ABC	50	2	0.96907	-0.0468	-0.08554	-0.01085	-0.07802
Dimensi	5	3	0.272125	0.897442	-0.00485	0.030227	0.106054
		4	0.883635	0.023276	0.175041	0.157743	0.290834
No. Test	1	5	0.764405	-0.03163	0.042561	-0.53341	-0.12314
		6	0.55012	0.060092	0.302484	0.311103	-0.19616
Waktu		7	0.227469	0.143667	0.773056	0.217214	-0.47301
Mulai	17:45	8	0.337508	0.133567	0.545404	-0.09208	0.703403
Selesai	18:07	9	0.849221	-0.04368	-0.19827	0.414572	0.040136
		10	0.457424	0.210936	0.787289	0.107563	0.17455
Fitness = 1231.164320							

**Tabel Lampiran A 10 Klasifikasi Uji Coba dengan SN=50,  
Iterasi=50, Percobaan ke-1**

Kelas Benchmark	Cluster											Total
	1	8	10	9	5	7	4	6	2	3		
PG	377	258	243	205	184	121	126	59	107	101		1781
Hu	184	153	121	121	116	78	77	37	67	56		1010
H&F	178	118	130	90	76	74	47	32	50	45		840
R&L	86	71	60	48	42	48	45	15	26	26		467
F/F/R	66	51	55	40	28	24	17	21	26	23		351
Fin	26	19	27	21	19	19	15	7	8	15		176
Car	21	16	19	13	11	15	13	2	8	8		126
E/T	22	9	15	9	8	2	6	10	4	4		89
TM/O	13	17	12	7	7	9	8	4	7	3		87
Phi	14	11	14	5	3	9	9	4	3	12		84
Total	987	723	696	559	494	399	363	191	306	293		

**Tabel Lampiran A 11 Confusion Matrix Uji Coba dengan  
SN=50, Iterasi=50, Percobaan ke-1**

Kelas Benchmark	TP	FP	FN	TN	Total
'Personal Growth'	377	610	1404	2620	5011
'Humor'	153	570	857	3431	5011
'Health & Fitness'	130	566	710	3605	5011
'Recreation & Leisure'	48	511	419	4033	5011
'Family/Friends/Relationships'	28	466	323	4194	5011
'Finance'	19	380	157	4455	5011
'Career'	13	350	113	4535	5011
'Education/Training'	10	181	79	4741	5011
'Time Management/Organization'	7	299	80	4625	5011
'Philanthropic'	12	281	72	4646	5011

**Tabel Lampiran A 12 Perhitungan Akurasi Uji Coba  
dengan SN=50, Iterasi=50, Percobaan ke-1**

Kelas Benchmark	Precision	Recall	Rand Measure
'Personal Growth'	0.381965552	0.211678832	0.598084215
'Humor'	0.211618257	0.151485149	0.715226502
'Health & Fitness'	0.186781609	0.154761905	0.745360208
'Recreation & Leisure'	0.085867621	0.102783726	0.814408302
'Family/Friends/Relationships'	0.056680162	0.07977208	0.842546398
'Finance'	0.047619048	0.107954545	0.892835761
'Career'	0.035812672	0.103174603	0.907603273
'Education/Training'	0.052356021	0.112359551	0.948114149
'Time Management/Organization'	0.022875817	0.08045977	0.924366394
'Philanthropic'	0.040955631	0.142857143	0.929554979
Rata-rata	<b>0.112253239</b>	<b>0.12472873</b>	<b>0.831810018</b>

**Tabel Lampiran A 13 Hasil Akhir Uji Coba  
dengan SN=20, Iterasi=25**

<b>Nilai</b>	<b>Test 1</b>	<b>Test 2</b>	<b>Test 3</b>	<b>Rata-rata</b>
<b>Fitness</b>	1223.598	1223.164	1223.948	1223.57
<b>Precision</b>	11.03%	11.05%	10.62%	10.90%
<b>Recall</b>	11.43%	12.86%	11.06%	11.78%
<b>Rand Measure</b>	83.14%	83.15%	83.31%	83.20%

**Tabel Lampiran A 14 Hasil Akhir Uji Coba  
dengan SN=30, Iterasi=25**

<b>Nilai</b>	<b>Test 1</b>	<b>Test 2</b>	<b>Test 3</b>	<b>Rata-rata</b>
<b>Fitness</b>	1230.63	1219.983	1222.321	1224.311
<b>Precision</b>	11.55%	11.55%	11.05%	11.38%
<b>Recall</b>	12.30%	12.70%	12.19%	12.40%
<b>Rand Measure</b>	82.97%	83.19%	83.14%	83.10%

**Tabel Lampiran A 15 Hasil Akhir Uji Coba  
dengan SN=50, Iterasi=50**

<b>Nilai</b>	<b>Test 1</b>	<b>Test 2</b>	<b>Test 3</b>	<b>Average</b>
<b>Fitness</b>	1231.164	1218.319	1227.064	1225.516
<b>Precision</b>	11.22%	11.05%	10.90%	11.06%
<b>Recall</b>	12.47%	12.28%	12.17%	12.31%
<b>Rand Measure</b>	83.18%	83.12%	83.10%	83.13%

**Tabel Lampiran A 16 Hasil Akhir Uji Coba  
dengan SN=100, Iterasi=100**

<b>Nilai</b>	<b>Test 1</b>	<b>Test 2</b>	<b>Test 3</b>	<b>Rata-rata</b>
<b>Fitness</b>	1220.909	1216.684	1222.198	1219.93
<b>Precision</b>	11.21%	11.10%	10.98%	11.10%
<b>Recall</b>	12.23%	12.37%	12.30%	12.30%
<b>Rand Measure</b>	82.99%	83.15%	83.10%	83.08%

## LAMPIRAN B

### SAMPEL TAHAPAN PROSES UJI COBA

Pada bagian ini dilampirkan hasil pengujian untuk setiap langkah demi langkah proses yang dijalankan pada program. Pengujian untuk bagian ini dijalankan dengan skenario khusus menggunakan jumlah sumber makanan (SN) = 5, dan iterasi ABC = 3. Jumlah dimensi dan banyak *cluster* yang dibentuk mengikuti pengujian standar, yaitu dimensi = 5 dan *cluster* = 10. Pengujian ini hanya bertujuan untuk memberikan sampel kinerja program secara rinci, dan tidak digunakan sebagai acuan hasil akhir pengerjaan Tugas Akhir .

➤ **Contoh data mentah**

- "#NewYearsResolution :: Read more books, No scrolling FB/checking email b4 breakfast, stay dedicated to PT/yoga to squash my achin' back!"
- #NewYearsResolution Finally master @ZJ10 's part of Kitchen Sink
- (5009 data lainnya)

➤ **Data setelah melalui proses *preprocessing***

- Read books scroll FB check email b breakfast stay dedic PT yoga squash achin back
- Final master part Kitchen Sink
- (5009 data lainnya)

- Data setelah melalui proses ekstraksi fitur TF-IDF dan reduksi dimensi LSA

**Tabel Lampiran B 1 Data Setelah TF-IDF dan LSA**

Data ke-	Fitur / Atribut				
	1	2	3	4	5
1	0.486183	0.166308	0.857417	0.014658	-0.02426
2	0.55712	0.233729	0.682879	0.267012	0.31204
3	0.248735	0.155937	0.553534	0.083673	0.774863
(4-5008)	...	...	...	...	...
5009	0.332303	0.07324	0.702993	-0.08519	0.618672
5010	0.303035	0.037658	0.548202	-0.23499	0.742298
5011	0.513774	0.110884	0.840271	0.086115	0.101343

- Inisialisasi sumber makanan

**Tabel Lampiran B 2 Hasil Inisialisasi**

SN	Centroid	Fitur				
		1	2	3	4	5
1	1	0.186855	0.122277	0.831815	0.262008	-0.43539
	2	0.298558	0.203539	0.877849	0.236521	0.207062
	3	0.309319	0.159049	0.560535	0.153764	0.735651
	4	0.417947	0.307361	0.673108	0.049911	0.524675
	5	0.480165	0.214151	0.79742	0.016175	0.295705
	6	0.511915	0.190891	0.772924	0.321396	0.028234
	7	0.754154	-0.03165	0.267034	0.436281	-0.41061
	8	0.837218	-0.10773	-0.17511	-0.45029	-0.23245
	9	0.888051	-0.08666	-0.2417	0.380017	0.032006
	10	0.984302	-0.06689	0.051912	0.103987	-0.11475



2	1	0.989701	-0.10401	-0.02644	-0.03893	-0.08636
	2	0.776988	-0.08182	0.133959	-0.59315	0.140808
	3	0.87315	-0.05078	0.105086	0.197476	0.430106
	4	0.293565	0.073461	0.533928	-0.13608	0.777707
	5	0.182872	0.119349	0.807037	0.145437	-0.52901
	6	0.399327	0.14642	0.853683	0.214683	0.210323
	7	0.464389	0.200995	0.654298	0.300756	0.474747
	8	0.42589	0.215262	0.831449	-0.26034	0.114884
	9	0.482679	0.280242	0.760742	0.101509	0.315361
	10	0.612619	0.741915	-0.27128	-0.0112	-0.02328
3	1	0.948016	-0.12462	-0.24035	-0.06049	-0.15591
	2	0.794852	-0.09889	-0.12579	-0.53136	-0.24548
	3	0.191812	0.96524	-0.06173	0.013111	0.165943
	4	0.989073	0.025532	0.109728	0.044121	0.084238
	5	0.783355	0.017191	0.346965	0.443095	-0.26333
	6	0.233394	0.142025	0.550356	0.089862	0.783829
	7	0.615263	0.112388	0.659699	0.188613	0.371541
	8	0.435477	0.267828	0.69822	0.289365	0.409126
	9	0.130914	0.094105	0.749072	0.209209	-0.60756
	10	0.299993	0.393721	0.807812	0.147824	0.283858
4	1	0.86875	-0.06422	0.11452	-0.45179	0.154658
	2	0.673384	-0.00463	0.4141	-0.21347	0.574007
	3	0.543866	0.161795	0.717355	-0.09697	0.392468
	4	0.550151	0.175529	0.619828	0.048258	0.529158
	5	0.970863	0.003256	0.130553	0.095928	0.176544
	6	0.387974	0.206865	0.632618	0.178276	0.612124
	7	0.399306	0.169816	0.832587	0.20233	0.278529
	8	0.643284	0.101392	0.709403	0.263561	-0.05648
	9	0.90469	-0.0731	-0.18156	0.37841	0.005946
	10	0.374768	0.178231	0.750555	0.428498	-0.28432

5	1	0.173356	0.125875	0.802371	0.227198	-0.50861
	2	0.758268	0.31981	-0.16456	-0.51144	-0.18465
	3	0.901344	-0.12162	-0.28642	0.294352	-0.06407
	4	0.746711	-0.50861	0.632618	0.550151	-0.09697
	5	0.536169	0.263235	0.779598	0.073474	0.173372
	6	0.348667	0.185809	0.778521	-0.38232	0.302727
	7	0.405257	0.240356	0.746711	-0.24578	0.400013
	8	0.847291	-0.04307	0.210566	-0.19023	0.446898
	9	0.375591	0.197545	0.65188	-0.01355	0.628312
	10	0.383792	0.043394	0.534212	-0.23221	0.715206

➤ **Optimasi inisialisasi**

**Tabel Lampiran B 3 Hasil Optimasi Inisialisasi**

SN	Centroid	Fitur					Nilai Fitness
		1	2	3	4	5	
1	1	0.228421	0.144979	0.772793	0.216013	-0.47229	1229.385
	2	0.448616	0.213311	0.650129	0.220005	0.481506	
	3	0.335527	0.129527	0.546464	-0.09205	0.705497	
	4	0.240282	0.784346	-0.00242	0.02268	0.08993	
	5	0.451906	0.219408	0.784549	0.111494	0.187615	
	6	0.882644	0.025072	0.173847	0.15729	0.297827	
	7	0.721601	0.074648	0.394529	0.376978	-0.23459	
	8	0.766511	-0.03432	0.041941	-0.53282	-0.12396	
	9	0.848255	-0.04187	-0.19579	0.415478	0.042517	
	10	0.969366	-0.04271	-0.08985	-0.01075	-0.07089	
2	1	0.880455	-0.04766	-0.15544	0.29117	-0.03035	1278.406
	2	0.712312	-0.03944	0.016198	-0.48337	-0.13178	
	3	0.886491	0.030211	0.167642	0.173026	0.22296	
	4	0.326237	0.122931	0.536162	-0.08326	0.726159	
	5	0.254789	0.144688	0.751019	0.230379	-0.47069	
	6	0.434883	0.210911	0.788113	0.235575	0.164496	
	7	0.458161	0.198324	0.617394	0.27512	0.495495	
	8	0.532839	0.203216	0.663292	-0.31285	0.143545	
	9	0.450441	0.238666	0.711914	0.038839	0.434225	
	10	0.270998	0.899725	-0.0051	0.031378	0.103109	

<b>3</b>	<b>1</b>	0.910619	-0.05458	-0.18738	0.241913	-0.02216	<b>1270.73</b>
	<b>2</b>	0.773972	-0.04389	0.020126	-0.52823	-0.14425	
	<b>3</b>	0.270417	0.900047	-0.00362	0.028937	0.104136	
	<b>4</b>	0.902503	0.021791	0.156803	0.137892	0.244884	
	<b>5</b>	0.683824	0.04751	0.161153	0.561904	-0.0154	
	<b>6</b>	0.329999	0.124814	0.541049	-0.08837	0.718146	
	<b>7</b>	0.407265	0.159175	0.49748	-0.251	0.138228	
	<b>8</b>	0.450097	0.215289	0.650067	0.195024	0.491255	
	<b>9</b>	0.252347	0.145038	0.756284	0.22623	-0.46904	
	<b>10</b>	0.442802	0.214834	0.786329	0.197784	0.18686	
<b>4</b>	<b>1</b>	0.772749	-0.04281	0.020707	-0.53275	-0.14058	<b>1248.449</b>
	<b>2</b>	0.328746	0.125105	0.541473	-0.08485	0.719018	
	<b>3</b>	0.523118	0.220641	0.685133	-0.21808	0.230729	
	<b>4</b>	0.462729	0.208496	0.63357	0.212635	0.490577	
	<b>5</b>	0.937148	-0.01567	0.033389	0.057603	0.106353	
	<b>6</b>	0.272251	0.898018	-0.00337	0.028789	0.104148	
	<b>7</b>	0.425951	0.215629	0.793051	0.217359	0.195575	
	<b>8</b>	0.575009	0.055771	0.299337	0.298915	-0.18338	
	<b>9</b>	0.849792	-0.03959	-0.18789	0.411856	0.049396	
	<b>10</b>	0.22646	0.145711	0.772235	0.220711	-0.47696	
<b>5</b>	<b>1</b>	0.232391	0.148911	0.76449	0.237926	-0.46875	<b>1272.248</b>
	<b>2</b>	0.781143	-0.05289	-0.01598	-0.55017	-0.10615	
	<b>3</b>	0.880292	-0.04724	-0.15375	0.29272	-0.0292	
	<b>4</b>	0.23922	0.784882	-0.00356	0.023834	0.090206	
	<b>5</b>	0.435921	0.213562	0.784444	0.231106	0.182076	
	<b>6</b>	0.6927	0.053374	0.439591	-0.17959	-0.41612	
	<b>7</b>	0.517015	0.22771	0.688008	-0.16754	0.294411	
	<b>8</b>	0.885192	0.030196	0.166643	0.174034	0.232444	
	<b>9</b>	0.445791	0.208918	0.641341	0.220415	0.498974	
	<b>10</b>	0.324778	0.124204	0.539442	-0.08637	0.723868	

➤ **Fase *Employed Bee* untuk iterasi ke-1**

**Tabel Lampiran B 4 Hasil Fase *Employed Bee*  
untuk iterasi ke-1**

SN	Centroid	Fitur					Nilai Fitness
		1	2	3	4	5	
1	Sumber makanan Sebelum K-Means						
	1	0.074413	-0.64474	-0.12717	0.806205	-1.31622	
	2	0.756382	-0.4356	-0.12212	0.585321	0.910374	
	3	0.747957	0.581772	0.692719	0.850165	1.086073	
	4	0.389035	0.193112	0.023945	-0.28709	0.806868	
	5	0.55027	1.218393	0.494995	-0.06706	0.207367	
	6	0.764812	-0.7983	-0.25954	1.048692	0.033305	
	7	0.451289	1.007999	0.24856	0.133764	-0.35434	
	8	1.644227	0.374547	-0.57026	-0.06239	0.350611	
	9	1.114983	0.619564	-0.22323	-0.46629	0.594185	
	10	1.75533	-0.59042	-0.16426	0.61616	-0.41958	
	Sesudah K-Means						1229.834
	1	0.44867	0.220074	0.786751	0.108474	0.19032	
	2	0.882978	0.025049	0.172791	0.156872	0.298212	
	3	0.44868	0.213317	0.649962	0.219679	0.481871	
	4	0.335469	0.129046	0.545723	-0.0926	0.706206	
	5	0.240282	0.784346	-0.00242	0.02268	0.08993	
	6	0.848255	-0.04187	-0.19579	0.415478	0.042517	
	7	0.228576	0.144817	0.772638	0.214924	-0.47246	
	8	0.968718	-0.04288	-0.08813	-0.00804	-0.07327	
	9	0.765854	-0.03467	0.042286	-0.531	-0.12659	
	10	0.715363	0.084673	0.417453	0.38199	-0.21017	

2	Sumber makanan Sebelum K-Means						1294.9
	1	0.108329	0.119033	0.654216	-0.31785	-0.04551	
	2	-0.17356	-0.77937	-0.82003	-1.22015	-0.53664	
	3	0.496343	0.262137	0.905537	0.674916	0.507238	
	4	0.407873	0.811642	0.646325	-0.86841	1.681981	
	5	0.387619	-0.17431	1.703787	1.018104	0.408675	
	6	0.272434	0.054297	1.578159	0.039231	0.102014	
	7	1.085424	0.340634	-0.0542	-0.48896	0.653278	
	8	0.791343	0.783812	-0.21033	-0.48406	0.857916	
	9	0.859892	0.044554	0.824416	-0.07697	0.852539	
	10	-0.33316	0.164805	-0.51499	-0.46032	0.192929	
	Sesudah K-Means						
	1	0.2394	0.131144	0.685591	0.205586	-0.42147	
	2	0.445137	0.214379	0.65561	0.207809	0.485874	
	3	0.458758	0.217574	0.779338	0.123325	0.180218	
	4	0.586234	0.085546	0.019846	0.680817	0.233998	
	5	0.896678	0.02422	0.171025	0.153773	0.228714	
	6	0.902778	-0.0513	-0.15341	0.253433	-0.04542	
	7	0.766637	-0.03372	0.041302	-0.52804	-0.12755	
	8	0.3896	0.826997	0.115563	0.047284	0.208408	
	9	0.335417	0.123118	0.547136	-0.09646	0.707753	
	10	0.137387	0.970623	-0.11157	-0.01944	0.005666	

3	Sumber makanan Sebelum K-Means						1310.472
	1	1.690986	-0.48965	-0.30618	0.442703	-0.7341	
	2	0.228807	0.217503	0.624646	-0.09121	-1.02171	
	3	1.1947	0.365441	-0.29411	0.724602	-0.72117	
	4	0.521978	-0.26443	-0.12675	0.456322	0.040242	
	5	0.82954	0.57003	0.758781	-0.19234	0.492608	
	6	-0.60297	-0.8172	0.387118	-0.37962	0.534838	
	7	0.167857	-0.51526	-0.21431	-0.98343	-0.39371	
	8	0.423479	-0.72265	-0.19638	-0.16256	-0.16596	
	9	-0.45951	0.998628	1.411846	-0.73438	-0.1226	
	10	0.221511	0.700251	1.093895	-0.16139	1.067463	
	Sesudah K-Means						
	1	0.135171	0.970766	-0.11692	-0.02147	0.002707	
	2	0.25866	0.144452	0.750687	0.238537	-0.46063	
	3	0.881316	0.018055	0.177906	0.183833	0.217896	
	4	0.888522	-0.08101	-0.15666	0.285436	-0.03331	
	5	0.45257	0.217229	0.684871	0.238209	0.400918	
	6	0.757719	0.485345	-0.14624	0.233243	0.057733	
	7	0.418434	0.187251	0.708359	-0.03673	0.121214	
	8	0.770852	-0.0372	0.031324	-0.52599	-0.13148	
	9	0.318053	0.859187	0.163223	0.044805	0.245157	
	10	0.344459	0.132911	0.552908	-0.06365	0.699524	

4	Sumber makanan Sebelum K-Means						1275.119
	1	1.732621	0.285567	0.153312	-0.66815	0.020955	
	2	-0.25572	0.567514	0.274799	-0.2425	0.585828	
	3	0.128554	0.205663	1.017814	-0.33267	-0.00301	
	4	0.185041	-0.68707	1.610249	-0.22975	1.059935	
	5	0.540505	0.045515	0.723321	0.974463	0.582869	
	6	0.118375	0.281378	0.481628	0.702983	-0.77336	
	7	0.88225	0.783082	0.065081	0.334596	-0.77859	
	8	1.075339	-0.29094	0.185581	1.2138	0.079826	
	9	-0.01392	0.30545	0.673342	1.272588	0.95084	
	10	-0.2185	0.601176	1.176641	0.878043	-0.23578	
	Sesudah K-Means						
	1	0.777034	-0.0491	-0.00673	-0.55399	-0.09411	
	2	0.335145	0.128926	0.545448	-0.09617	0.706701	
	3	0.440097	0.222313	0.787618	0.132652	0.210588	
	4	0.631432	0.098472	0.497324	-0.32826	-0.28206	
	5	0.460013	0.208228	0.634671	0.214998	0.49194	
	6	0.221492	0.14776	0.774733	0.230152	-0.47339	
	7	0.23922	0.784882	-0.00356	0.023834	0.090206	
	8	0.85467	-0.0409	-0.14749	0.376506	0.109784	
	9	0.956297	-0.01447	0.008271	0.012361	0.016912	
	10	0.734141	0.09347	0.408123	0.367739	-0.12424	

5	Sumber makanan Sebelum K-Means						1254.688
	1	0.781001	0.889299	1.5954	-0.22847	-0.82375	
	2	1.576234	0.007731	-0.75325	-0.696	0.19839	
	3	1.73272	0.583795	-1.07645	-0.00779	0.925471	
	4	-0.49568	0.505738	0.694023	-0.59863	0.51523	
	5	0.101213	-0.63948	1.740268	0.857046	0.553629	
	6	1.664447	-0.30634	0.244241	-0.96084	-1.26509	
	7	-0.07311	-0.08088	0.191167	0.496511	0.610789	
	8	1.11845	-0.05658	-0.39117	0.22671	0.922608	
	9	0.099385	-0.53808	1.619507	0.513257	1.096525	
	10	0.922098	-0.05015	-0.29342	-0.85118	0.032964	
	Sesudah K-Means						
	1	2.28E-01	1.45E-01	7.73E-01	2.17E-01	-4.73E-01	
	2	4.50E-01	2.19E-01	7.89E-01	9.65E-02	1.85E-01	
	3	7.07E-01	8.39E-02	4.01E-01	4.02E-01	-1.85E-01	
	4	-4.44E-16	0.00E+00	-2.22E-16	2.78E-17	1.94E-16	
	5	2.72E-01	8.98E-01	-2.54E-03	2.94E-02	1.04E-01	
	6	4.50E-01	2.14E-01	6.54E-01	2.24E-01	4.72E-01	
	7	3.38E-01	1.31E-01	5.46E-01	-8.62E-02	7.04E-01	
	8	8.84E-01	-5.18E-02	-1.89E-01	2.90E-01	-4.33E-03	
	9	9.05E-01	2.40E-02	1.39E-01	1.36E-01	2.37E-01	
	10	7.67E-01	-3.30E-02	3.98E-02	-5.27E-01	-1.27E-01	

➤ ***Greedy Selection setelah Fase Employed Bee***

Untuk tahapan selanjutnya, sumber makanan yang akan digunakan hanya sumber terbaik dari seluruh tahapan sebelumnya, yang dicari menggunakan algoritma *greedy selection*. Jumlah sumber makanan yang diambil adalah 5. Sumber makanan yang diambil pada tahap ini, urut dari sumber dengan nilai *fitness* terbaik adalah sumber dari fase optimasi ke-1, *employed* ke-1, optimasi ke-4, *employed* ke-5, dan optimasi ke-4.

➤ ***Fase Onlooker Bee untuk iterasi ke-1***

**Tabel Lampiran B 5 Hasil Fase Onlooker Bee  
untuk iterasi ke-1**

SN	Centroid	Fitur					Nilai Fitness
		1	2	3	4	5	
1	Nilai Random : 0.943867						
	Sumber Makanan Terpilih : 5						
	Sumber makanan Sebelum K-Means						
	1	0.154584	0.350249	-0.71961	-0.17802	-0.64563	
	2	-0.02918	-0.91637	-0.01417	-0.04826	-0.77849	
	3	0.714984	1.402093	0.337192	-0.37522	-0.04307	
	4	1.677228	0.480384	1.020849	-0.70855	0.192126	
	5	1.656615	-0.13866	-0.76391	-0.39599	0.780983	
	6	1.056568	-0.39402	0.912324	0.734379	-0.26552	
	7	-0.39464	0.475609	-0.42227	-0.2068	1.111785	
	8	0.571594	0.54131	1.448194	0.10466	-0.4632	
	9	-0.62187	-0.48013	1.277898	0.100056	0.310729	
	10	0.655881	0.393458	1.658189	0.907565	-0.36814	
	Sesudah K-Means						1254.688
	1	0.91151	-0.05492	-0.18622	0.241668	-0.02221	
	2	0.673571	0.072937	0.46663	-0.22146	-0.35525	
	3	0.269753	0.901157	-0.00208	0.02323	0.10311	
	4	0.776831	-0.0465	-0.00604	-0.55244	-0.09421	
	5	0.900884	0.022188	0.159019	0.144698	0.243318	
	6	0.534698	0.045996	0.109438	0.473006	-0.00172	
	7	0.336803	0.128329	0.544637	-0.09899	0.703817	
	8	0.232154	0.146632	0.767377	0.23831	-0.46787	
	9	0.446418	0.214638	0.650575	0.209774	0.489176	
	10	0.455075	0.219329	0.780318	0.134088	0.18848	

2	Nilai Random : 0.500514					1264.959	
	Sumber Makanan Terpilih : 3						
	Sumber makanan Sebelum K-Means						
	1	0.187089	0.578415	-0.72403	-0.06761		-0.90122
	2	0.229945	-0.75815	-0.02662	0.706171		0.282704
	3	0.972365	-0.71348	0.905398	0.039288		0.467437
	4	1.030293	0.769495	0.32083	-0.49145		0.253015
	5	0.338207	0.948655	-0.08394	0.402271		-0.13407
	6	0.546663	1.384233	0.101009	-0.16192		-0.87829
	7	0.367073	0.509469	0.874296	-0.72449		-0.19454
	8	0.989742	-0.43525	-0.44421	-0.2364		0.029361
	9	0.981841	0.048635	0.6895	0.662121		0.65094
	10	0.643463	0.103159	0.766608	1.20979		-0.46529
	Sesudah K-Means						
	1	0.291607	0.196181	0.575287	0.068969		0.703669
	2	0.887987	0.02889	0.161535	0.167958		0.230425
	3	0.381128	0.072981	0.522325	-0.19832		0.700414
	4	0.77378	-0.0415	0.02568	-0.52537		-0.13688
	5	0.23922	0.784882	-0.00356	0.023834		0.090206
	6	0.462749	0.216128	0.651636	0.202477		0.477959
	7	0.473888	0.214225	0.768999	-0.12301		0.166606
	8	0.879447	-0.04715	-0.15266	0.293625		-0.03103
	9	0.449977	0.213005	0.757678	0.28021		0.196772
	10	0.255384	0.144176	0.750662	0.231796		-0.4699

3	Nilai Random : 0.413531					1324.327	
	Sumber Makanan Terpilih : 3						
	Sumber makanan Sebelum K-Means						
	1	0.097018	-0.96413	-0.76502	0.34821		0.712628
	2	0.755787	0.735853	0.806711	0.826677		1.350469
	3	-0.10597	0.998221	0.82127	0.33479		0.303603
	4	-0.4352	0.08529	1.216964	0.083341		0.390547
	5	1.916462	-0.38297	-0.04006	0.332493		0.362933
	6	-0.17668	1.097763	0.0191	0.722733		-0.03498
	7	-0.19232	-0.20384	1.440061	-0.58366		-0.11054
	8	-0.18705	-0.53668	-0.22253	-0.03331		-0.13224
	9	1.827685	-0.66791	0.6926	1.391776		-0.93742
	10	0.492336	-0.23031	1.268624	-0.11677		-0.92358
	Sesudah K-Means						
	1	0.79662	0.013456	0.018668	0.391176		0.297713
	2	0.878038	0.041347	0.277605	0.16682		0.025903
	3	0.447684	0.214607	0.652605	0.211172		0.483787
	4	0.33655	0.128998	0.545139	-0.09459		0.706168
	5	0.913518	-0.06508	-0.19393	0.233821		-0.03068
	6	0.267867	0.902152	-0.0021	0.029341		0.104781
	7	0.452366	0.2184	0.784021	0.122569		0.182971
	8	0.706023	-0.06566	0.036844	-0.48797		-0.11454
	9	0.720138	0.380375	0.025639	-0.39594		-0.09186
	10	0.252744	0.145401	0.755619	0.226788		-0.46307



4	Nilai Random : 0.592311						
	Sumber Makanan Terpilih : 3						
	Sumber makanan Sebelum K-Means						
	1	-0.17061	-0.14778	-0.70268	0.311703	-0.6995	
	2	-0.31697	0.550833	0.691465	-0.79072	1.549164	
	3	0.49813	0.43182	0.020227	-0.7007	0.274209	
	4	1.31374	1.062671	0.910903	-0.11571	0.858598	
	5	0.985011	0.900062	-0.70596	0.204725	-0.85173	
	6	1.257012	1.202406	-0.67517	-0.19078	1.033818	
	7	-0.07944	-0.22262	-0.09123	-0.03198	-0.1692	
	8	1.167571	-0.795	-0.5561	0.144665	-0.49686	
	9	0.160658	-0.66648	0.670496	-0.24294	-0.09615	
	10	0.997687	0.055355	0.395586	1.134961	-0.53975	
	Sesudah K-Means					1277.873	
	1	0.391058	0.822722	0.1069	0.062411		0.209116
	2	0.437332	0.216885	0.78789	0.211244		0.181489
	3	0.536317	0.192757	0.643543	-0.21754		0.323547
	4	0.312472	0.123728	0.546367	-0.07511		0.730631
	5	0.137387	0.970623	-0.11157	-0.01944		0.005666
	6	0.450058	0.210272	0.644442	0.217625		0.491672
	7	0.712313	-0.03942	0.021001	-0.48218		-0.13252
	8	0.883309	-0.04762	-0.16085	0.285603		-0.02969
	9	0.258381	0.144714	0.751849	0.223179		-0.46527
	10	0.878673	0.029615	0.17222	0.192743		0.217038
5	Nilai Random : 0.146689						
	Sumber Makanan Terpilih : 1						
	Sumber makanan Sebelum K-Means						
	1	-0.45227	0.572357	1.303553	1.138078	-0.93771	
	2	0.880861	-0.50576	0.515639	0.93518	0.409677	
	3	0.565768	-0.42881	-0.04454	-0.72415	-0.08813	
	4	0.195954	1.2246	0.385694	0.614822	0.614152	
	5	1.432679	-0.68045	0.245986	-0.79699	0.909357	
	6	1.06135	0.522159	-0.63119	-0.19643	-0.59973	
	7	1.518204	0.364126	0.33028	1.353063	-0.2123	
	8	0.037902	-0.41974	-0.73723	-1.39	-0.23214	
	9	0.150721	-0.80516	-0.05132	0.517386	0.565685	
	10	1.882967	-0.02593	0.173215	0.619086	-0.29035	
	Sesudah K-Means					1291.395	
	1	0.227767	0.145571	0.772641	0.213848		-0.47513
	2	0.457548	0.207649	0.633536	0.222423		0.493855
	3	0.773099	-0.04418	0.021196	-0.53298		-0.14044
	4	0.23922	0.784882	-0.00356	0.023834		0.090206
	5	0.419835	0.213815	0.797415	0.221311		0.19362
	6	0.848966	-0.03952	-0.18725	0.412604		0.049847
	7	0.733452	0.077384	0.389509	0.362392		-0.21055
	8	0.530629	0.223498	0.67972	-0.16942		0.277107
	9	0.325573	0.125037	0.540286	-0.08366		0.722313
	10	0.937832	-0.01684	0.02938	0.056462		0.107322

➤ ***Greedy Selection setelah Fase Onlooker Bee***

Sumber makanan yang diambil dengan algoritma greedy selection pada tahap ini adalah sumber dari dari fase optimasi ke-1, *employed* ke-1, optimasi ke-4, *employed* ke-5, dan *onlooker* ke-1.

➤ ***Fase Scout Bee untuk iterasi ke-1***

Tidak terdapat sumber makanan yang jumlah kemunculannya melebihi variabel parameter *limit* yang berjumlah 100, sehingga pada fase ini tidak dilakukan proses apapun.

➤ ***Sumber makanan terbaik setelah iterasi ke-1***

Karena fase *scout bee* tidak menghasilkan sumber makanan baru, maka sumber makanan terbaik untuk keseluruhan iterasi ABC yang pertama sama dengan sumber makanan yang diambil untuk proses *greedy selection* setelah fase *onlooker bee*.

➤ ***Iterasi ABC selanjutnya***

Iterasi selanjutnya menggunakan tahapan yang serupa dengan iterasi pertama. Hasil yang telah dilampirkan sebelumnya dirasa telah memenuhi tujuan sebagai sampel, sehingga untuk iterasi kedua dan ketiga tidak dilampirkan lagi.

➤ **Sumber makanan terbaik setelah 3 iterasi**

**Tabel Lampiran B 6 Hasil Akhir  
Sumber Makanan Terbaik**

	Fitur					Nilai
Centroid	1	2	3	4	5	Fitness
<b>1</b>	0.228421	0.144979	0.772793	0.216013	-0.47229	<b>1229.385</b>
<b>2</b>	0.448616	0.213311	0.650129	0.220005	0.481506	
<b>3</b>	0.335527	0.129527	0.546464	-0.09205	0.705497	
<b>4</b>	0.240282	0.784346	-0.00242	0.02268	0.08993	
<b>5</b>	0.451906	0.219408	0.784549	0.111494	0.187615	
<b>6</b>	0.882644	0.025072	0.173847	0.15729	0.297827	
<b>7</b>	0.721601	0.074648	0.394529	0.376978	-0.23459	
<b>8</b>	0.766511	-0.03432	0.041941	-0.53282	-0.12396	
<b>9</b>	0.848255	-0.04187	-0.19579	0.415478	0.042517	
<b>10</b>	0.969366	-0.04271	-0.08985	-0.01075	-0.07089	

Demikian telah dilampirkan sampel yang diambil demi memperjelas proses yang dikerjakan dalam alur program. Perhitungan akurasi tidak dilakukan karena sampel yang diambil disini tidak digunakan sebagai data hasil akhir keseluruhan pengerjaan Tugas Akhir ini.

*(Halaman ini sengaja dikosongkan)*

## **BIODATA PENULIS**



Rizki Novianto, biasa dipanggil dengan Rizki, lahir di Surabaya, 28 November 1991. Penulis merupakan anak kedua dari tiga bersaudara. Penulis menempuh pendidikan formal dimulai dari TK Dharma Wanita Rungkut (1995-1997), SDN Kendangsari III Surabaya (1997-2003), SMP Negeri 35 Surabaya (2003-2006), SMA Negeri 17 Surabaya (2006-2009) dan sebagai mahasiswa Teknik

Informatika Institut Teknologi Sepuluh Nopember Surabaya (2009-2017).

Di Teknik Informatika ITS, penulis mengambil bidang minat Komputasi Cerdas dan Visualisasi (KCV). Penulis dapat dihubungi melalui e-mail: [alten.nth@gmail.com](mailto:alten.nth@gmail.com)